TIBCO iWay® Service Manager

User's Guide

Version 8.0 and Higher
March 2021
DN3502112.0321

Contents

Chapter 1

Introducing iWay Service Manager

iWay Service Manager (iSM) is an Enterprise Service Bus (ESB) product that enables you to create, compose, and manage services – whether deployed as web services, APIs, or services accessible through other commonly-used and industry standard interfaces. iSM lays the foundation for a quickly deployable and easily maintainable Service-Oriented Architecture (SOA) or Event-Driven Architecture (EDA) for your enterprise.

In this chapter:

iWay Service Manager Functions

iWay Service Manager (iSM) is lightweight, high performance, general purpose, and highly configurable. It includes design-time graphical tools to help you create sophisticated integration services without the need, in many cases, for any custom programming.

Functions of iSM and its associated tools include:

❑ Creating metadata from target applications and data sources.

❑ Transforming messages and mapping between disparate interfaces.

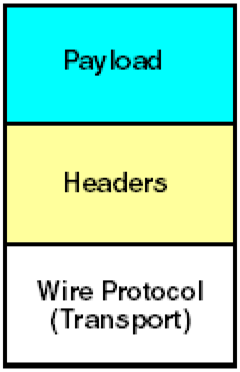
❑ Designing and managing stateless processes.

❑ Composing channels using a flexible modular agent framework.

It also provides common development and run-time services for your adapters (services) such as browser-based console support, multithreading management, trace logging, and archiving and restoring of deployed configurations.

Understanding iWay Channels

iWay Service Manager (iSM) is designed to host channels and their constituent components. In the simplest case, a channel enables a message to be received from one adapter and emitted via another adapter. In general, a message arriving at the server has three main parts as shown below. The diagram shows the stack organized with the more physical layers at the bottom and more abstract, “business-oriented” layers at the top.

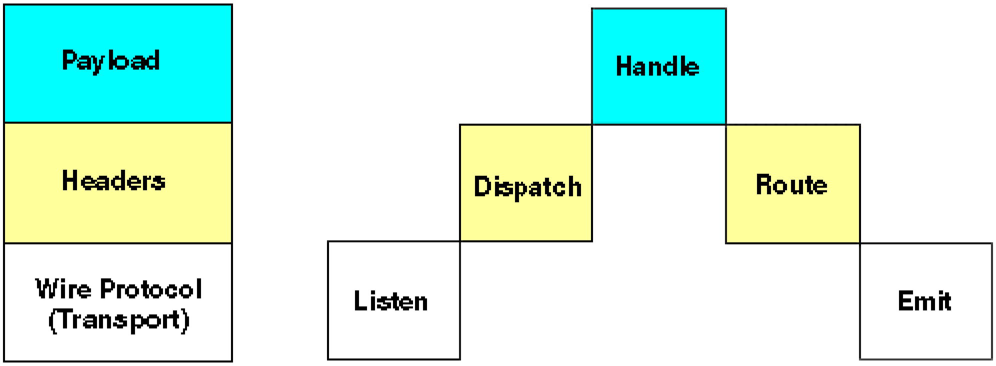


Payload. Regardless of the transport protocol on which the message arrives, the payload is the data representing the content of the message. This payload may be in native (flat), XML, or JSON format. Examples of such messages are EDI, SOAP, or an API request. The channel can be configured to parse the incoming payload into an internal format, to leave the incoming payload in its current format (no parsing) or to use supplied or redeveloped preparsers to convert a unique payload format into an internal format.

Message headers. Headers contain contextual information about the payload or protocol interactions, such as encoding information, content type, and so on. In a HTTP transport, these are the HTTP headers. In a MQ transport, these are part of the Message Queuing Message Descriptor (MQMD). Regardless of the transport, the headers are made available to the application in a common manner.

Transport. This is the transmission method by which the message reaches (or is emitted by) iSM. iSM supports a wide variety of transport protocols, such as HTTP(S), many queuing mechanisms (MQ, JMS, Rabbit, etc.), FTP, files, and so on.

An iWay adapter handles all three layers of the message. iSM makes it easy to configure each of these independently. The following diagram shows how these layers map onto functions in iSM.



Conceptually, the inbound message moves up the stack from its physical protocol to where its logic and semantics are handled, and then back down the stack to be emitted. This may occur using a single type of protocol adapter or with the inbound and outbound sides using different protocols.

The transport is handled by a listener (for incoming messages where iSM is acting as a server) and an emitter (for outgoing messages where iSM is acting as a client). Note that iSM frequently behaves as a client and a server within a given installation. In almost all cases, the transport components supplied with iSM will be sufficient to construct your channels since the product’s support of industry standard protocols is complete.

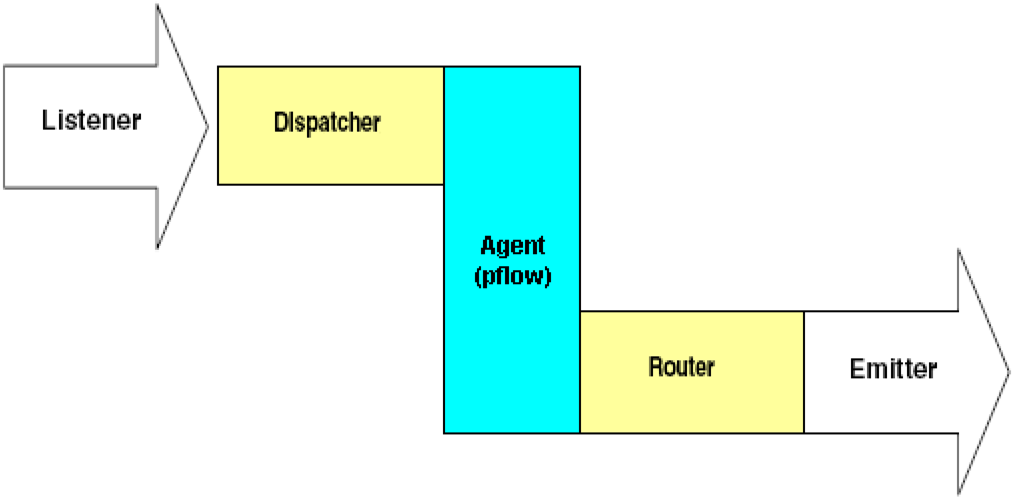
The headers are handled by dispatching and routing. This is the middleware level where much of the added value of iSM resides and where the iWay preparser exits are executed (see *Preparsers* in *Developing for Your Requirements* on page 21. Although a complete set of exits is provided with iWay, you may be required to create a new exit for security or for formatting a message into XML.

Although a non-XML message can be routed through the system and acted upon by specific exits, iSM operates most effectively with XML documents. If your incoming message is not in XML and if one of the standard preparsers provided by iWay does not suit your requirements, you may want to write a preparser exit. For more information, see the *iWay Service Manager Programmer’s Guide*.

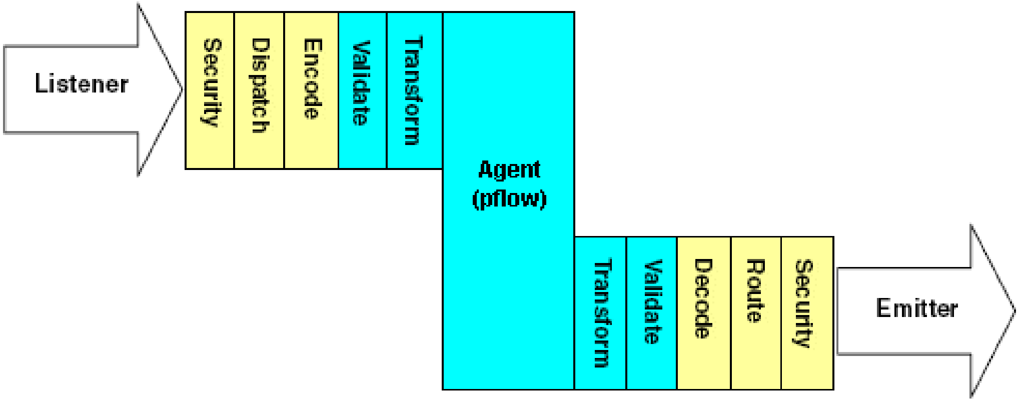
Finally, the message is handled by the application level. This is where iWay business agents are executed, and the level where integration applications are written. Although many business agents are provided with the product, you may write a custom business agent for specific programming requirements. For more information on the business agents that are supplied with iSM, see the *iWay Service Manager Component and Functional Language Reference Guide*.

In most cases, you need only assemble the business agents already provided into a process flow using iWay Integration Tools (iIT) Designer. For more information, see the *iWay Integration Tools Designer User's Guide*. Mapping one message type to another can be accomplished in the XML domain using iIT Transformer. Transformations are created graphically using this tool, and then published to iSM for run-time use. For more information, see the *iWay Integration Tools Transformer User's Guide*.

The following diagram illustrates the flow of the message or document through a simplified, single-directional channel.



The phases of working with the message or document can be further subdivided. Various exits are provided for dispatching and routing, and may be indicated for your application. If all the available exits are shown, the flow looks like the following diagram.

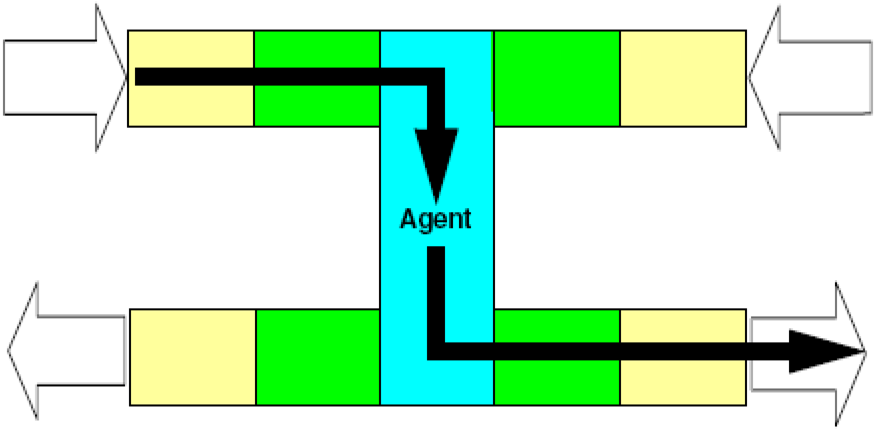


The first step after receiving the message can be to verify the security of the message, in order to ensure that the sender is authorized, check that the message was not changed, and decrypt parts of the message that were encrypted. iWay has security exits for many common security algorithms, including W3C Digital Signature, and PGP, among others. From the exit, you can also check credentials in an external security manager such as Netegrity, Active Directory, or an LDAP server.

Before their payloads can be handled, all messages must frequently be parsed into an internal format. iSM can automatically parse XML and JSON to internal form. However, process flows can handle flat documents as well. The dispatching step is called for non-parsed messages to encode their contents as required by the application. This is called a *preparser* because it occurs *before* the message is parsed from its unknown format into the an appropriate internal form. On the outbound side, the step is called a *preemitter*, and can encode an internal format such as an XML tree or a JSON object into a format appropriate for its recipient. iSM supports many non-standard XML or JSON formats for preparsing and preemitting, thereby eliminating the requirement to code a specific exit.

Finally, there is an exit to validate the contents of the payload. This validation can use iSM rules as described in the *iWay Service Manager Programmer's Guide*.

Because an integration application generally requires both an input and output, a typical iSM channel is constructed with at least two adapters. One adapter handles the incoming message and another handles the outgoing message, as shown in the following diagram.



The arrow on the diagram shows a one-way message. This adapter allows messages originating on the left to be delivered to the right. However, this basic configuration may support the following two scenarios:

- ❑ A **one-way message.** If the message is one-way, the listener does not wait for a response from the system that is the target of the emit before completing the listen operation.

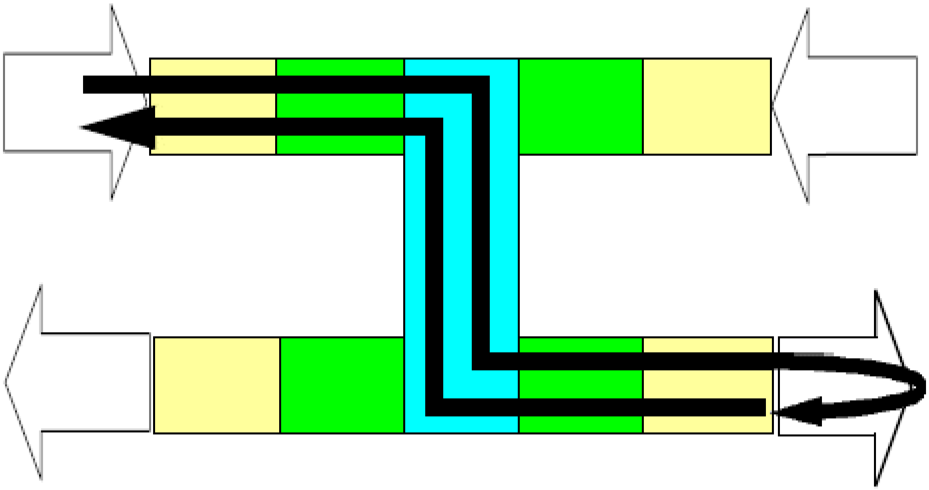
  For an asynchronous protocol listener, such as MQSeries or the iSM internal listener, no acknowledgment is sent.

  For a synchronous protocol listener, such as HTTP, an acknowledgment is sent back.

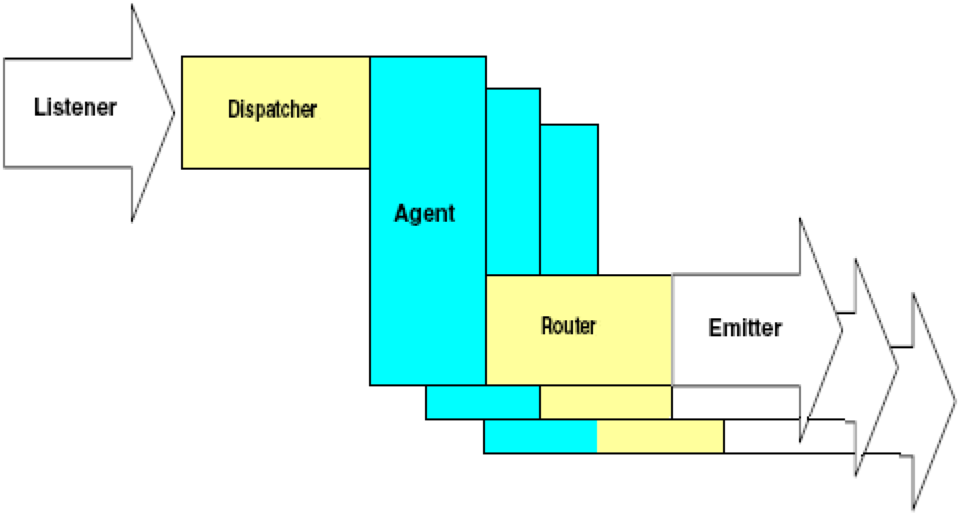
  Note: One-way messages are always asynchronous at the application level even if they use a synchronous protocol at the transport level.

- A **request-reply** where the reply is received synchronously at the emitter, and the listener holds the request until the reply document is made available and returned.

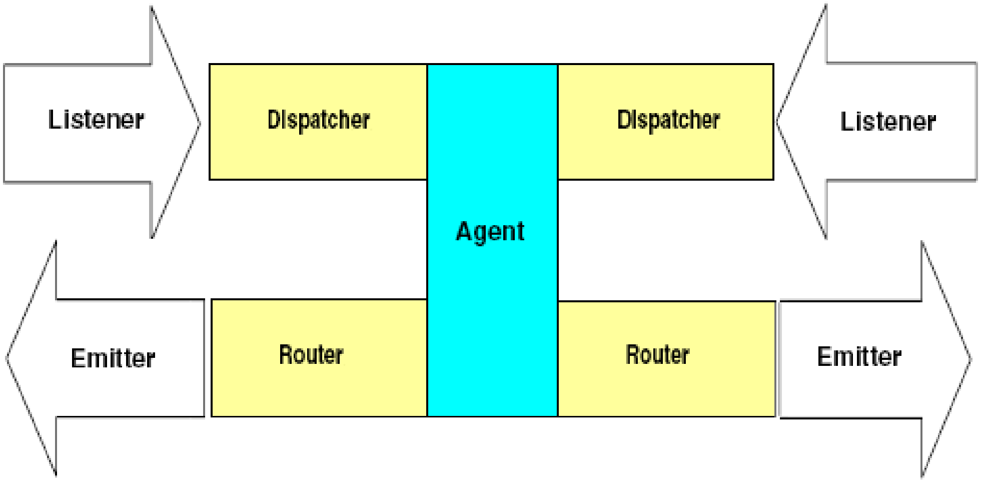
  A synchronous message, including a reply, is shown in the following diagram.



Although the previous diagram shows a single path through the channel, multiple routes, agents, and outlets can be defined in a channel. In that case, there can be multiple outputs for a single input, either to the same adapter or to different adapters as shown in the following diagram.



Channels can be bidirectional. The following diagram shows a bidirectional channel which has a listener and emitter on both sides. Messages can arrive from the right or the left and can be emitted on the left or the right, respectively.



To accomplish its mission, iSM executes a defined flow as each message travels from input to output. The flow is standard for all messages, but the execution steps and the processing at each step vary by message source and type. Each of the steps is performed by an exit, which is a routine invoked to perform the specific action at that step. This is the essence of the modular, *pluggable* framework, which is the source of the flexibility and simplicity of iSM.

In some respects, iSM can be thought of as simply a platform for the execution of exits. No document processing is performed by the server itself. Instead, the server assembles the appropriate exits to accomplish the business purpose of the document. In this model, building a channel is the selection and ordering of pre-built exit code. The existing exits can be supplemented by application-specific code as necessary to accomplish the interfacing task.

iSM protocol handlers can also be supplemented by new protocols tailored to specific application situations. These protocol handlers, called listeners and emitters, service the input channels and output ports that represent the iSM interface to the outside world. As mentioned earlier, the need for custom protocol adapters is rare given the adoption of standard protocols by most enterprises.

For more information on writing custom exits or protocol handlers, see the *iWay Service Manager Programmer's Guide*.

Developing for Your Requirements

iWay Service Manager (iSM) provides a wide range of business exits for constructing your custom channels. Exits apply in document parsing, output, validation, transformation, security, and agent execution. For more information, see the *iWay Service Manager Component and Functional Language Reference Guide*.

Exits are written in Java and can take advantage of standard, core iWay system services during their execution. Services such as document parsing, document transformation, tracing, and use of an internal object store are all made available when appropriate to the task of the exit.

Much of the standard functionality of the system is provided through iWay-supplied code run at the exit points, which may be replaced or supplemented by user-written code for special tasks. Full Javadoc support is available with the installation of the server.

The standard exit points are:

- **Document processing**

  These exits handle the actual payload and are called *business agents*. They are at the application and business level. Business agents execute after all input transformations and represent the turn-around point at which an input document becomes an output document. Business agents usually implement the business logic associated with the document. Business agents are the basis of process flows.

- **Preparsers**

  Most messages in XML or JSON format can be automatically parsed by iSM. Preparser exits are available to manipulate the input before it becomes a document in an internal iWay format (XML or a JSON object). Preparser exits can directly produce an internal format such as an XML tree, or can prepare the incoming message to be parsed by a subsequent iWay parser. For example, a HIPAA input document in EDI format that is to be processed through the system might be converted into XML format.

  Preparsers can be chained so that the output of one preparser can be passed to a following preparser. Input to the first preparser is a byte stream, and output is in a string form. Subsequent preparsers in the chain accept strings and emit strings. Following the last preparser, the engine normally parses the string as an internal document. A common use of this approach it to *split* large incoming documents into smaller portions that can be processed one at a time. This type of a preparser is called a *splitting preparser* in iSM documents.

- **Preemitters**

These exits are available to process an output document before it is emitted by iSM. The standard emit mechanism in iSM is to *flatten* the input format into a payload that is suitable for transport. In the event that this simple flattening is not appropriate for the destination, preemitters can be employed to change the internal format to an appropriate external format. For example, a business message might be transformed into a flat EDI message format.

Preemitters can also be chained. The first preemitter accepts a document as input and emits a byte stream. Subsequent preemitters accept the byte stream output from the first preemitter and emit a byte stream. Specialized encryption can also be implemented by preemitters.

❑ **Rules**

These exits receive control through the rules system which checks nodes of the incoming and outgoing document for validity. The rules system is discussed in the *iWay Service Manager Programmer's Guide*.

Usually the exits are not invoked directly by package name, but by alias. An alias is a short name associated with the exit and stored in the system repository.

Document Life Cycle

The concept of a channel has been introduced to assist in the construction of message flows in iWay Service Manager (iSM). A channel serves as a container for all your components that simplifies the integration design process and improves organization, versioning, and troubleshooting.

Each message passes through a channel, and in many applications through multiple channels. Messages can be XML or non-XML, based on the demands of the application. Channels can be developed in meaningful groups using iWay Integration Tools (iIT), which is beyond the scope of this manual. Channels are named in iSM as *"groupname":"inletname":listenername"* as configured in iIT. One or more channels constitute an application, which can be configured and deployed as a single entity.

A channel contains the following elements that must be configured and associated with that channel:

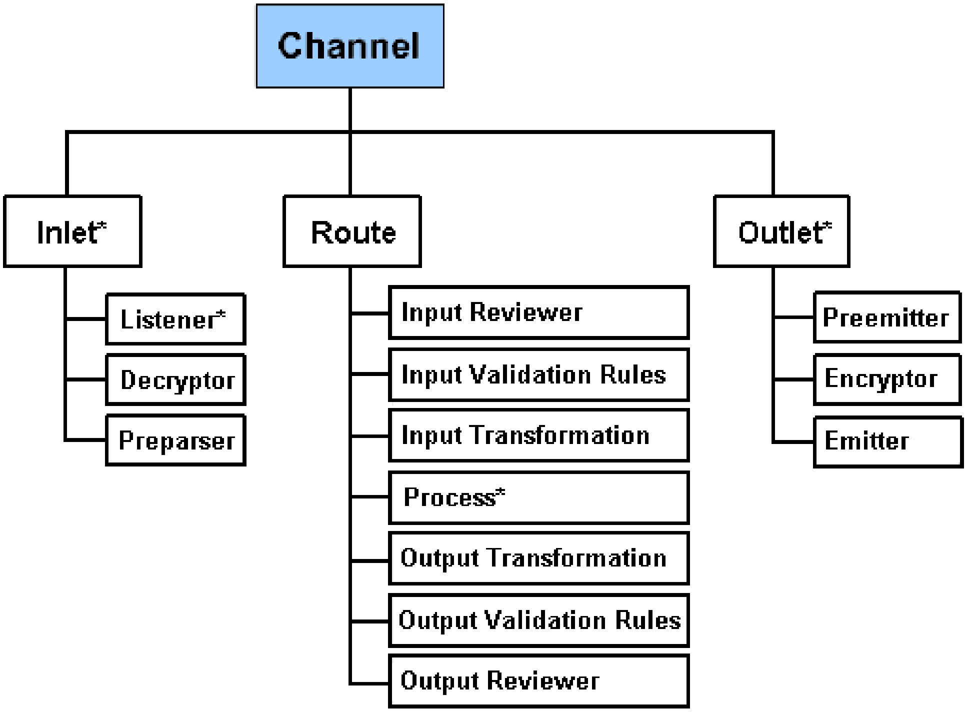
❏ **Inlet.** Defines the point at which a message entry point enters the channel, and optionally, some preliminary steps in the message processing to prepare for subsequent stages. Each channel must contain an inlet. In the inlet, the message is received on a supported protocol and may be decrypted and/or preparsed and then assigned to an appropriate route for further processing. Inlets contain listeners, decryptors, and preparsers. Generally, decryption using standard security algorithms are simply configured, and decryption is simply built into iSM.

❏ **Route** - Defines the path a message takes through a channel. Each channel must contain a reference to one or more routes. Along each route, transformations and other forms of business logic are applied to the message. Routes contain transforms, processes, and emitters. Conditional logic, expressed using iWay Functional Language (iFL) is used to select the appropriate route for each message. For more information on using iFL, see the *iWay Functional Language Reference Guide*.

❏ **Outlet.** Defines how a message leaves a channel at the end of the process. Each channel must contain one or more outlets. In the outlet, the message may be transformed and/or encrypted, and finally emitted by the system. Each message may be sent to multiple outlets of a channel. Outlets contain preemitters, encryptors, and emitters. Outlet selection can be performed through iFL expressions.

The components of a channel are shown in the following diagram.

Note: The components indicated with an asterisk (*) must be configured for a channel (Inlet, Listener, Process, or Outlet). The other exits are optional.



Document Flow Through a Channel

The document travels through the channel components as follows:

- ❑ **Inlet Phase:**
  - ❑ **Listener.** Protocol handler that receives input for a channel from a configured source. iSM offers a wide variety of standard protocols, include non-blocking HTTP, File input, AS2, SQL, and so on.
  - ❑ **Decryptor.** Applies a decryption algorithm, which must be supplied in a custom module.
  - ❑ **Preparser.** Transforms an external format to a suitable internal format. You can choose from several preparsers that are supplied with iSM or use a custom preparser.

- **Route Phase:**
  - **Optional Input Reviewer.** Analyzes the document. This agent is typically used to extract and handle headers.
  - **Input Validation Rules.** Applies validation using the rules validation engine. Rules are provided when HIPAA, HL7, and SWIFT adapters are installed, or you can write custom rules for input documents.
  - **Input Transformation.** Applies iWay transformations or XSLT transformations to input documents received by a listener. Transformations are more typically handled in the process flow, but in some cases route transformations are used to make diverse inputs common so as to use a standard process flow.
  - **Process.** Executes the business logic of an application. Process flows are developed using iWay Integration Tools (iIT), and are composed of services that operate on the message, control elements such as conditional tests, and the paths from one service to the next.

    Processes can take advantage of multiprocessor and multithread logic. You can choose from a wide variety of services provided by iWay or you can use custom-written agents that access third-party applications and systems. You can use multiple services in series or in parallel.

    Common services include transformations, data enrichment, exchange of the message with other channels running on the same application, another application potentially on another machine, integration with other iWay products, such as such as iWay Data Quality Server (DQS), and emitting messages to external targets. Processes can be configured to be transactional.
  - **Output Transformation.** Applies iWay transformations or XSLT transformations to all XML outbound response documents.
  - **Output Validation Rules.** Applies validation using the rules validation engine. Rules are supplied for HIPAA, HL7, and SWIFT adapters, or they may be custom written for output documents.
  - **Output Reviewer.** Analyzes the document. This agent usually adds headers.
- **Outlet Phase:**
  - **Preemitter.** Transforms internal formats to a suitable external format for transport.
  - **Encryptor.** Applies an encryption algorithm, which must be supplied in a custom module.

- **Emitter.** Protocol handler that drives the output of a channel to a configured endpoint.

Note: Parsing is implied in channel execution and occurs between the inlet and route phases. This step may be disabled on your channel, depending on your requirements.

Applying Business Logic to Messages

iWay Service Manager (iSM) executes business logic based on the message that arrives and the protocol on which it is received. To accomplish a task in iSM, you must apply the appropriate business logic at the appropriate time. The business logic that is applied relates to both the message being processed and the processing stage. The task of iSM is to apply the configured business logic at the appropriate stage of message processing.

Configuration determines the business logic to apply to a message. Using the configuration, you can refer to built-in logic elements, or you can supply your own.

To apply business logic, iSM must know:

- How to locate the logic, regardless of whether it is built-in or supplied externally.
- The stage at which it is to be associated with the message.
- The logic elements to apply in each case.

The major stages in the life of a message or document are:

- On arrival, before it is parsed into an internal format (XML or JSON).
- After parsing, before execution of a business function.
- Execution of a business function.
- Before it is emitted to a specific end point on a protocol.

Business logic is associated with each of these stages.

You can define a business logic element from the Registry section in the iSM Administration Console, or when you add it to a channel or listener. After a business logic element is defined, it can be applied to messages in a specific message flow. Business logic can also execute iWay Designer process flows.

Business logic, in the form of document-content based routing, may be performed by a listener or from within a process flow. For more information, see the *iWay Service Manager Programmer's Guide*.

Process flows are usually deployed on the server on which they will be executed. However, process flows can also be housed in an external/remote library and checked out for use as needed. This allows the process to be carried as a named exit that might, for example, be identified in an external data storage, such as iWay Trading Partner Manager (TPM). Similarly, transforms can also be carried in libraries and called by name. This includes iWay and XSLT transforms.

iWay Functional Language

iWay Service Manager (iSM) provides the iWay Functional Language (iFL), which can be used to configure parameters in iSM components. iFL can obtain information to be used for this configuration from:

❏ External sources (for example, property files or the system environment).

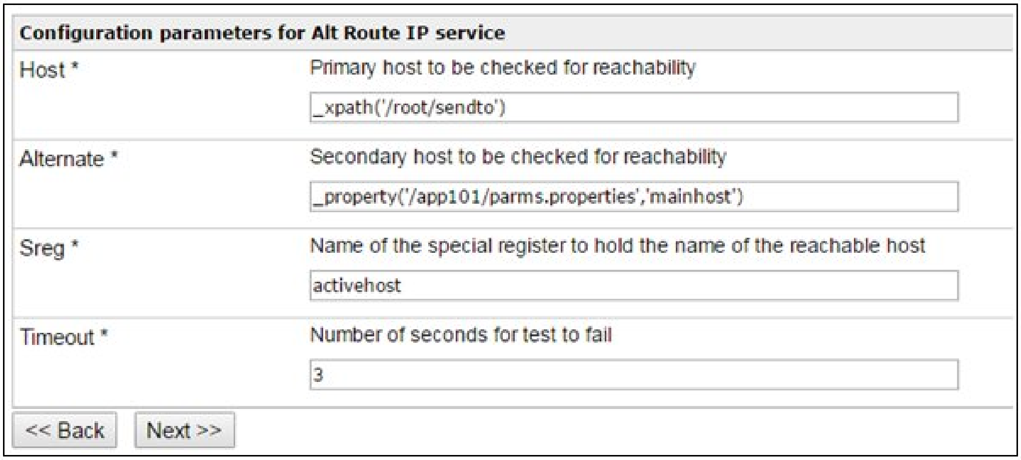
❏ The message context and characteristics (for example, MSMQ, HTTP headers, or message length).

❏ The message itself (for example, XPath or JsonPath, JsonPointer).

iFL also offers string processing, conditional expressions, and scoped variables.

For example, an application receives an XML document containing an address to send as a response. If the host cannot be reached, then a default alternate is stored in a property file. The Alternate Route IP Service (com.ibi.agents.XDAltRouteIP) checks to determine whether the primary host can be reached. If the primary host cannot be reached, then an alternate host is selected by this service. The alternate host name to be used is stored in a subsequent emit operation in a variable (called a Special Register or SREG). The subsequent emit service can access the selected host name using the following iFL _sreg() function:

```
_sreg('activehost')
```



Configuration parameters for Alt Route IP service

Host *	Primary host to be checked for reachability	_xpath('/root/sendto')
Alternate *	Secondary host to be checked for reachability	_property('/app101/parms.properties','mainhost')
Sreg *	Name of the special register to hold the name of the reachable host	activehost
Timeout *	Number of seconds for test to fail	3

If the incoming message was a JSON document, then the value for the Host parameter that is shown in this example might be:

```
_jsonptr('/root/sendto')
```

For more detailed information about using iFL and all of the functions that are available, see the *iWay Functional Language Reference Guide*.

Understanding iWay Service Manager Thread Management

iWay Service Manager (iSM) is engineered as a modern, multi-threaded server. It accepts messages on threaded inlets and processes many messages in parallel. Computers use threads as a means of allocating computational resources. Operating systems take responsibility for allocating those resources among threads. Although different operating systems may implement the heuristics of thread/resource allocation in different ways, the general purpose is to get as much work as possible through the application in a given period of time. The usual heuristic is to let one thread execute while another thread is awaiting completion of some slow task such as awaiting user input or the completion of an I/O event.

iSM is designed to maximize the work able to be performed on arriving messages to provide high throughput, or the number of messages that can be processed in a given period of time. Messages are isolated from each other during their processing, insuring that inadvertent interaction does not take place and that a failure of one message does not accidentally affect other messages.

In multiprocessor hardware, the threads are executed by separate CPUs, and the ability of the application to take advantage of multiple CPUs is referred to as scalability. iSM works to avoid the interlocks that reduce scalability in order to take maximum advantage of the resources available to its execution.

In iSM, some thread management is automatic; for example, the use of threads to watch for run-away process flows or to respond to console requests. These threads have minimal impact of the actual message execution and are not considered further in this section.

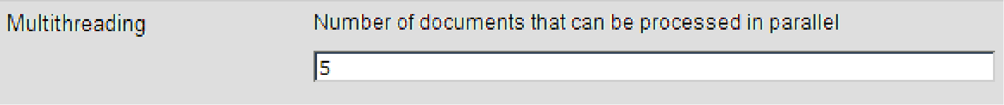
User configuration of listeners affects the use of threads in iSM. The following section explains how these settings work together and how they can be expected to affect the use of threads.

How Listeners Use Threads

Inlets are the portion of a channel that accepts and prepares messages for execution. In an inlet, the portion that accepts and prepares messages for execution is called a listener. Listeners implement the specific protocol needed to obtain the message.

Each listener executes in its own thread, and never shares resources with other listeners. The manager starts and stops listeners, and gathers and distributes statistics on their operation. Listeners are designed to be reasonably inactive; they mostly await events and prepare them for execution. The execution is delegated to a worker thread, which in iSM is called a worker. When you set the thread count for a listener, you are declaring how many workers that listener has to accept and process work. A message is only bound to a worker during its execution, from when it is ready to be processed until that processing is complete. Completion of process is not meant in a business sense – rather it means the handling on the message until the current operation is completed.

Because workers take time to start up and shut down, the server allocates to each listener a pre-initialized set of workers, called the listener's worker pool. The Multithreading field on the configuration console instructs iSM startup on how many workers are to be allocated to the pool. When a message arrives, the listener selects a worker from the pool, assigns the message to it, and then moves on to await the arrival of the next message.



Multithreading

Number of documents that can be processed in parallel

5

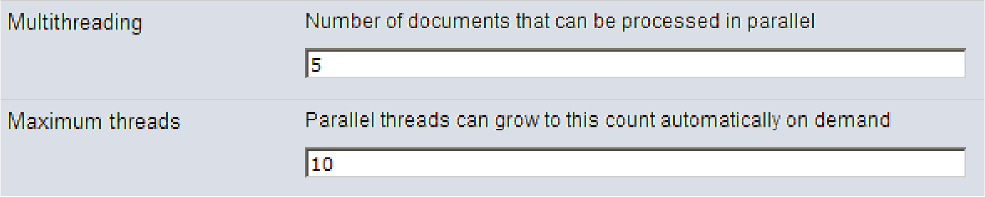
The consequence of the multithreading number varies from protocol to protocol. Protocols that hold queues, such as any of the queue listeners like MQ or internal, grab a message and request a worker for its execution. The listener waits until a worker is available and then hands off the work. If the number of workers is less than the number in the queue, then the messages wait in the queue until a worker is available for their execution. Given an expected arrival distribution and estimated time of service for a message, you can calculate how long a message will wait.

Maximum Listener Thread Settings

There are other listeners that cannot wait for any period of time. An example is TCP or HTTP. The listener must accept the message and pass it to a worker for execution or else the message gets lost.

If there are no workers available to handle the message, you can turn many of the listeners into queue listeners by writing the message onto an internal queue and then getting back to the sender with an "I got it" message. For example, this technique is used (although not with the internal listener) for AS2 – listeners take the message, return an MDN and then proceed to handle the message. In this way, the time for a worker to become available is minimized.

You may want to increase the number of workers running in parallel to handle the actual traffic. For example, in HTTP you might usually expect five simultaneous messages (five workers) but once in a while you need ten workers. To do this, set threads to 5 and max to 10. When the sixth message arrives, the worker pool "sees" that there are no workers available and creates a sixth worker to get the message. In this way, the message is not lost. If the sixth worker finishes, it goes back into the pool which now has six. However, because each worker takes resources, the system watches the pool and after some time, it destroys the sixth worker, since all you need is five. The extra was to handle the peak situation.



Multithreading	Number of documents that can be processed in parallel 5
Maximum threads	Parallel threads can grow to this count automatically on demand 10

Some listeners do not offer the ability to grow threads, and some do.

You cannot set the max threads to a number which is less than the multithreading count. The threading count is the number of waiting workers. They are pre-started to avoid setup time when messages arrive, so the size of the pool is the multi thread count. Do not confuse this with OS threads; it is not guaranteed that they map one to one, although in most JVMs they do.

The Internal Listener and its Affect on the Threading Model

The internal protocol is an iSM-managed queuing protocol that accepts messages from other channels for asynchronous execution. The main purposes of the internal protocol are:

❑ to offer the opportunity to change from one threading model to another.

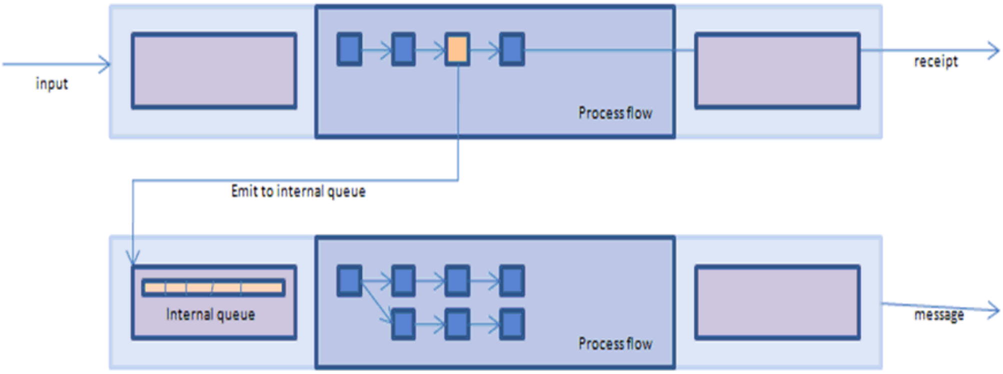
❑ to alleviate stress on another channel by executing work independently of the response to the sender.

❑ to break up (modularize) the application for simpler development and maintenance.

When the Internal Emit Service is used in a process flow, the message is immediately added to a named internal listener queue. For more information, see the *iWay Cross-Channel Services Guide*.

When the internal emit service sends the message to the internal queue, it immediately completes, passing the document out for any further processing. A common use of the internal listener is to have the first channel perform needed preliminary processing and then immediately send a receipt message (called an MDN for Message Delivery Notification) while the actual business processing proceeds without reference to the actual source. In the AS2 protocol, the logic in the internal listener channel could prepare an asynchronous notification of the result of the business processing. The recipient would need to correlate an asynchronously received result with the original message. Because the internal listener channel displays the original transaction ID, the asynchronous message can contain the needed information to assist in the correlation.

The diagram shows the link between two channels. The inlet and outlet handling is shown as purple boxes. The top channel process flow sends the message to the internal listener channel and then immediately proceeds to complete operation of the first stage.



The internal listener channel can operate on a different number of threads than the original recipient of the message. For example, it may be necessary to receive the messages on a single thread to preserve input message number tracking. However, once the message is determined to be in the correct sequence, the actual execution can proceed on multiple threads.

Alternatively, it may be required to receive on a large number of threads, but for a final stage of execution only one thread might be possible. This type of requirement can occur because of restrictions on a resource (for example, a fixed number of connections to an online system, such as SAP) or the need to control output sequence numbering in some manner.

The internal listener offers two forms of application flow throttling control:

- **Passivation**

  Passivation refers to sending instructions to other listeners (channels) to cease acquiring messages from their own input sources. The internal listener does this by allowing the configuration of a high and low mark.

  When the number of messages on the internal listeners queue exceeds the high mark, the passivation is sent to the identified channels. The internal listener continues to acquire messages from its queue, and when the number in the queue reaches the low mark, the identified channels are reactivated. Channels not passivated can continue to acquire messages for their processing and add messages to the internal queue.

- **Inhibition**

Inhibition can be configured for the channel. It causes the internal queue itself to reject attempts to add to the queue while the number of messages on the queue exceeds the high mark. The inhibition state is reset when the queue reaches the low mark. The flows attempting to add to the inhibited queue enter a pause state, and the add attempt may time out. In such a timeout case, a status document is passed down the timeout edge of the feeding flow, if available. Use of inhibition can have a cascading effect on the application, causing a general pause in processing while the internal channel catches up with the required work.

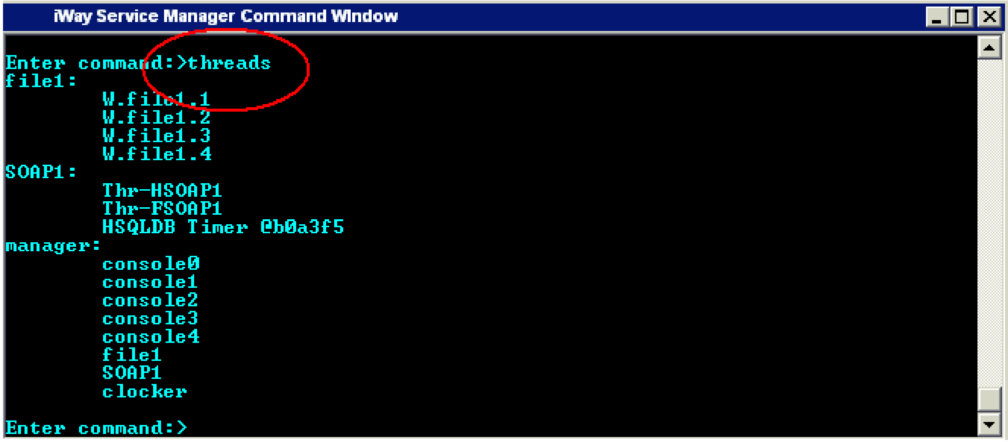
The internal channel also supports message priority. Usually, the message is placed on the queue at a middle priority. Using priorities allows the sender to reorder message execution through the channels as required. As a best practice, iWay recommends using priorities only if needed. Use priorities one up or one down from the offered value. Avoid using the highest priority value (9). Priorities are relative to other messages, and higher numbers do not necessarily result in faster processing.

The Ordered Listener

A specialized form of the internal listener is the ordered listener. This is a patented facility that automatically sorts and relates messages to ensure that the messages are presented to the listener in a desired order (time, lexical or numeric compare, or *application-specific*). In addition, an ordered listener ensures that the messages in any ordered group are processed sequentially while also processing multiple groups in parallel. This facility simplifies the development of applications that require identified processing orders.

Getting Thread Information

When operating iWay Service Manager (iSM) in a command window or Telnet console, many commands exist to assist in understanding (and possibly debugging) the current state of the server. One of the more useful commands is `threads`, as shown in the following image.



```
Enter command:>threads
file1:
        W.file1.1
        W.file1.2
        W.file1.3
        W.file1.4
SOAP1:
        Thr-HSOAP1
        Thr-FSOAP1
        HSQLDB Timer @b0a3f5
manager:
        console0
        console1
        console2
        console3
        console4
        file1
        SOAP1
        clocker

Enter command:>
```

In this particular display of an iSM instance with two listeners, SOAP1 and file1, you can see the thread running each protocol listener (the channel) and one thread for each message handler (worker) that it controls. There will be one thread for each currently available worker in the pool.

The SOAP listener's sub-threads include one for handling HTTP and one for file management. There is also a repository control thread that in this example is related to the HSQL data base.

The threads are controlled by the manager thread which is in charge of the server. The manager controls five console threads, two channels, and the server's heartbeat clock used to coordinate time-related issues such as detecting run-away process flows.

Best Practices

The flexibility of iWay's threading models enables the server to be tailored to a wide variety of application needs. While no set of suggestions are valid for all applications, some overarching guidelines should be considered as best practices.

1. Decide before you start configuring what performance you need. What constitutes success? Just going fast for its own sake takes up resources that might be better spent elsewhere.
2. Use as few threads as possible to keep up with the application's needs.

3. Sometimes fewer threads give better performance than many threads. For example, applications configured using only a few threads in reading from an IBM Websphere MQ queue have been clocked as completing work faster than when the listener is configured with many threads.
4. Try to keep the number of worker threads in active use to a small multiple of the number of available processors.
5. Process flows with long wait times built in, such as those that address external systems over communications lines, will need more threads that those that simply work in memory doing such things as transformations.
6. Don't be afraid to try various configurations. Measure as you go. And remember that threads are not the only resource that affects performance: consider memory availability and other processes running on the computer. And when measuring, remember that Java optimizes code after many transactions run through the server. Do not start to measure until you have run at least 1000 transactions.

Understanding iWay Service Manager Transaction Management

iWay Service Manager (iSM) manages transactions to ensure that multiple actions either *all complete* or *none complete*. This is standard transaction management for servers. iSM uses concentric transaction control, addressing the different requirements for a listener or a process flow. Naturally, actual transaction control is influenced by the protocol and external resources themselves, and so, this general discussion must always be considered in light of the environment.

Listener transactions (attempt to) ensure that messages are processed one time only, but are not lost. To this end, messages are read under transaction control. Completion of processing of the message, regardless of outcome, completes the transaction and the message is deleted from the source. For example, the message is picked from an MQ queue and then deleted at the end. On the other hand, if the server should fail during mid-processing, then the transaction does not complete and the message is made available by the original source for re-execution.

Once read, the message is processed by the channel itself. This includes the process flow. During the execution, specific services, such as SQL, can associate themselves with the transaction and receive notifications in a two-phase commit manner as the process flow ends.

The process flow itself can end in one of three ways:

❑ **Successfully.** In this case, any associated services receive a successful notification and can commit their work with any external resources with which they are associated.

❑ **General Failure.** The associated services receive notification and perform a rollback as needed.

- **Fail Service Execution.** The *Fail* service can be included in the process flow to signal a failure, which is handled as a general failure, but under program control.

Note that use of a *Catch* in the process flow provides error handling, and thus the outstanding failure is ignored. The process flow can use a *Fail* service to terminate the flow from within the *Catch* handling if desired.

The *Fail* service can also trigger a *Retry*, which for supporting protocols causes the original message to be resubmitted following a configured delay. This can be continued until the message succeeds or a configured retry limit is reached. If the retry limit is reached, then the message is considered to be permanently failed. However, this can be reset by configuring a failure event flow.

Chapter 2

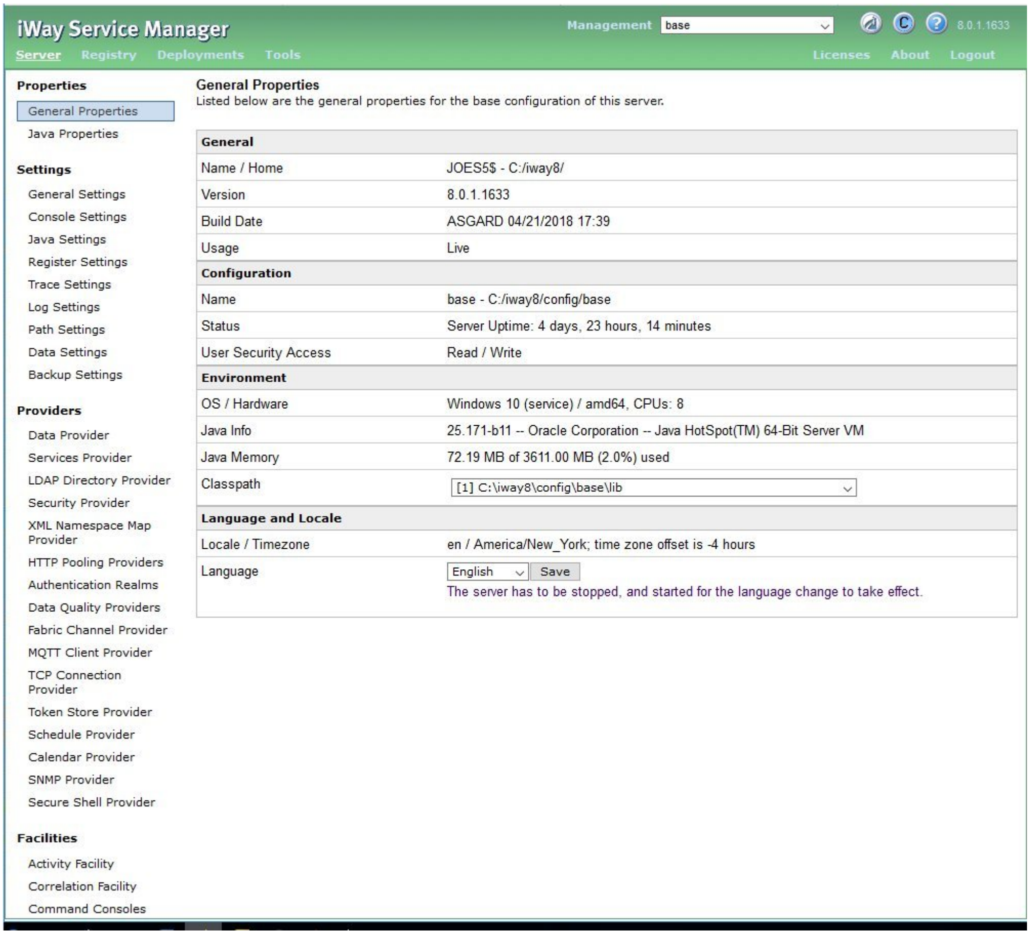
Introducing the iWay Service Manager Administration Console

The iWay Service Manager Administration Console provides access to all operating properties. To access the console, iWay Service Manager must be running.

In this chapter:

Console Home Window

The following image shows the default iWay Service Manager Administration Console open to the General Properties pane. There is a navigation menu across the top and a configuration menu in the left pane. The right pane displays information related to the selected option.



General Properties

Listed below are the general properties for the base configuration of this server.

General	
Name / Home	JOES5$ - C:/iway8/
Version	8.0.1.1633
Build Date	ASGARD 04/21/2018 17:39
Usage	Live
Configuration	
Name	base - C:/iway8/config/base
Status	Server Uptime: 4 days, 23 hours, 14 minutes
User Security Access	Read / Write
Environment	
OS / Hardware	Windows 10 (service) / amd64, CPUs: 8
Java Info	25.171-b11 -- Oracle Corporation -- Java HotSpot(TM) 64-Bit Server VM
Java Memory	72.19 MB of 3611.00 MB (2.0%) used
Classpath	[1] C:\iway8\config\base\lib
Language and Locale	
Locale / Timezone	en / America/New_York; time zone offset is -4 hours
Language	English Save The server has to be stopped, and started for the language change to take effect.

The contents of the right pane of the console change in accordance with the option selected in the left pane.

Main Menu

The navigation bar at the top of the console window contains the main menu. The main menu provides access to submenus, which open in the left pane, that give access to all iWay Service Manager functions. These menus can be used to configure, monitor, and diagnose channels, data sources, and display system properties.

The following table lists and describes the main menu options.

Option	Description
Server	Displays the iWay Service Manager General Properties pane. The submenus on the left enable you to configure server-level settings.
Registry	Displays the iWay Service Manager general registry properties. The registry is used to define and manage resources within the design-time repository of iWay Service Manager. This registry gives access to the configuration of resources such as adapters, emitters, listeners, services, and encryptors.
Deployments	Displays information related to the run-time configuration of the iWay Service Manager, including the settings for each channel and the current status of each channel.
Tools	Displays the iWay Service Manager log viewer, and gives access to the iWay Service Manager package management facilities. In addition, this pane allows you to configure access handlers for the iWay Trading Manager and the iWay Enterprise Index. These handlers control run-time access to the repository for these products.
Managed Servers	Clicking the Managed Servers link displays the Server Management pane, which allows you to add new server configurations and users. The drop-down list enables you to select which of your defined server configurations to manage.
iWay Software Technical Support	Launches a browser accessing the iWay Software Technical Support website, where you can open support cases, review open cases, and perform other maintenance and research tasks.

Option	Description
Check Website for Updates ✓	Provides current release and installation information and enables you to download updates.
C	Provides access to the Information Builders' Customer Community.
Help ?	Provides access to the Welcome pages, online documentation, a glossary, and Javadoc. For more information, see *iWay Service Manager Help Window* on page 41.
Restart Restart	Recycles iWay Service Manager to put into effect any new configuration changes that have been made. This does not restart the underlying JVM, so parameters at that level will not be reloaded.
Licenses Licenses	Displays information about your license file, including the iWay Software adapters you are authorized to run.
About About	Provides information about the installed software and general licensing information.

General Properties Page

By default, the general properties page is the first page seen upon logging in to the server. It includes the following categories of information:

General

- □ Logon ID and iWay Home Directory.
- □ Version.
- □ Build Date.
- □ Usage.

Configuration

- Configuration Name and working directory (where the iWay Service Manager configuration files are installed for the managed server currently in use).
- Configuration Status, which displays the server uptime in hours and minutes.
- User Security Access (the access rights of the user who logged on to the console).
    - Full access enables editing of the iWay Service Manager configuration.
    - Read-only access enables only viewing of the configuration.

Environment

- System Operating System and Hardware.
- Version of JVM (Java Virtual Machine).
- Java Memory Information.
- Java Classpath entries.

Language and Locale

- Time Zone and differential from Greenwich Mean Time (GMT).

  All times used in iWay Service Manager are in GMT.

  The differential from local time also appears.

- Language. The language used by the iWay Service Manager Administration Console.

  The default language is English. For instructions on how to change the language, see *How to Select the Language Displayed by the Console* on page 49.

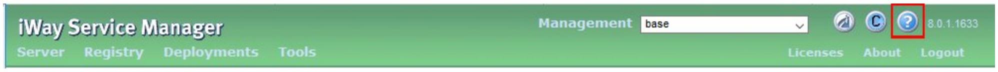
iWay Service Manager Help Window

The iWay Service Manager Administration Console provides a Help window that provides links to various online help topics.

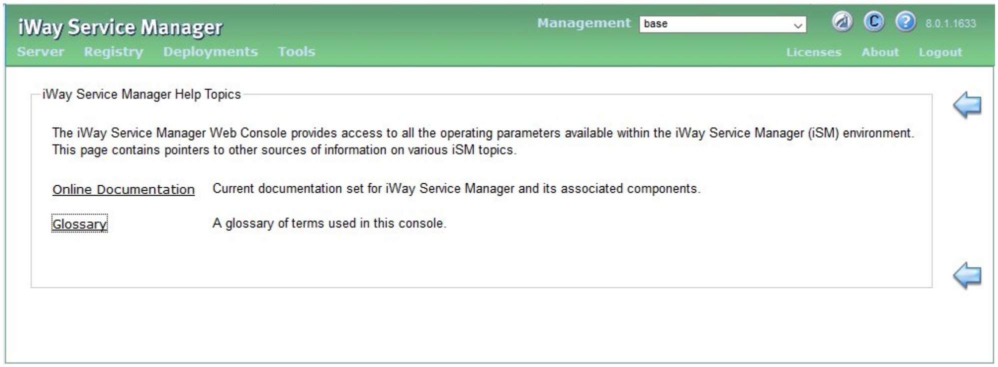
Procedure: How to Open the iWay Service Manager Help Window

To open the iWay Service Manager Help window:

1. Click the *Help* icon in the navigation bar at the top of the console window.



The iWay Service Manager Help Topics window opens.



iWay Service Manager Help Topics

The iWay Service Manager Web Console provides access to all the operating parameters available within the iWay Service Manager (iSM) environment. This page contains pointers to other sources of information on various iSM topics.

Online Documentation Current documentation set for iWay Service Manager and its associated components.

Glossary A glossary of terms used in this console.

This Help window provides the following topics:

- **Welcome** - Opens the iWay Service Manager Welcome window, which includes links to a section on getting started, tutorials, samples, and new features. For more information, see *iWay Service Manager Welcome Window* on page 42.
- **Online Documentation** - Opens the Documentation window, which allows you to browse the current documentation set for iWay Service Manager and its associated components.
- **Glossary** - Opens the Glossary window, which lists and describes some common terms that are used when working with iWay Service Manager.
- **iWay Functional Language** - Opens the Function and Conditional Expressions window, which lists and describes the various functions used by the iWay Functional Language.
- **Console Commands** - Opens the Console Commands window, which lists and describes the various commands that can be used to help manage the server.
- **Javadoc** - Opens the Javadoc window, which provides an overview of the business exits in iWay Service Manager and a link to the Javadoc API. To access Javadoc, you must first build and deploy the Javadoc sample.

2. Click one of the links to access the corresponding topic.

iWay Service Manager Welcome Window

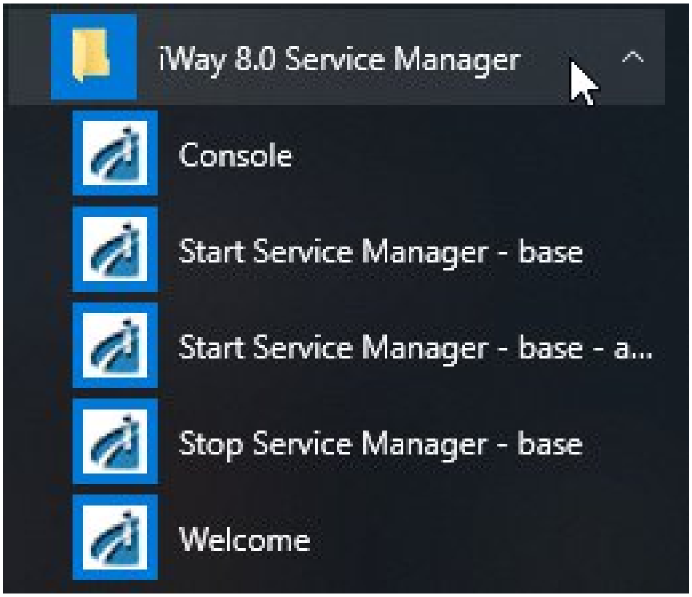
The iWay Service Manager Administration Console provides a Welcome window to help you get started and achieve a better understanding of the features and capabilities iWay Service Manager has to offer.

The following section describes how to access the iWay Service Manager Welcome window and provides information about the various topics that are available.

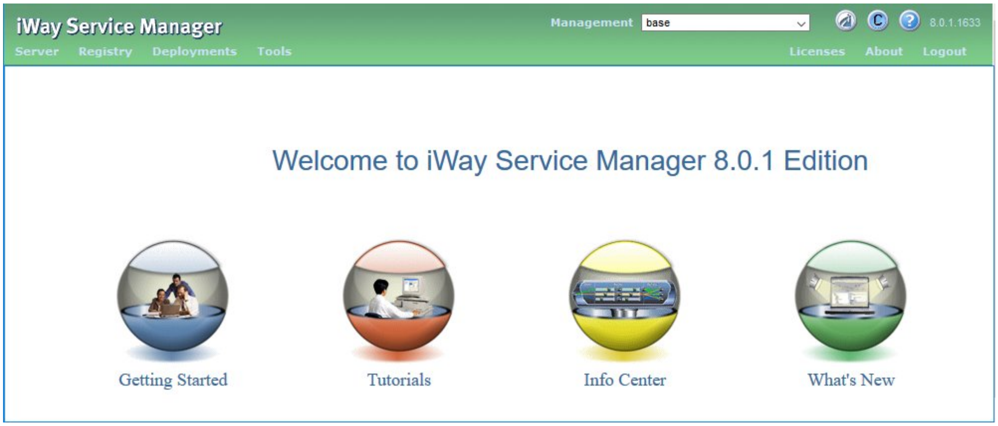
Procedure: How to Open the iWay Service Manager Welcome Window

To open the iWay Service Manager Welcome window:

1. Select *Welcome* from the Start menu.


iWay 8.0 Service Manager
Console
Start Service Manager - base
Start Service Manager - base - a...
Stop Service Manager - base
Welcome

The iWay Service Manager Welcome window opens.



Welcome to iWay Service Manager 8.0.1 Edition

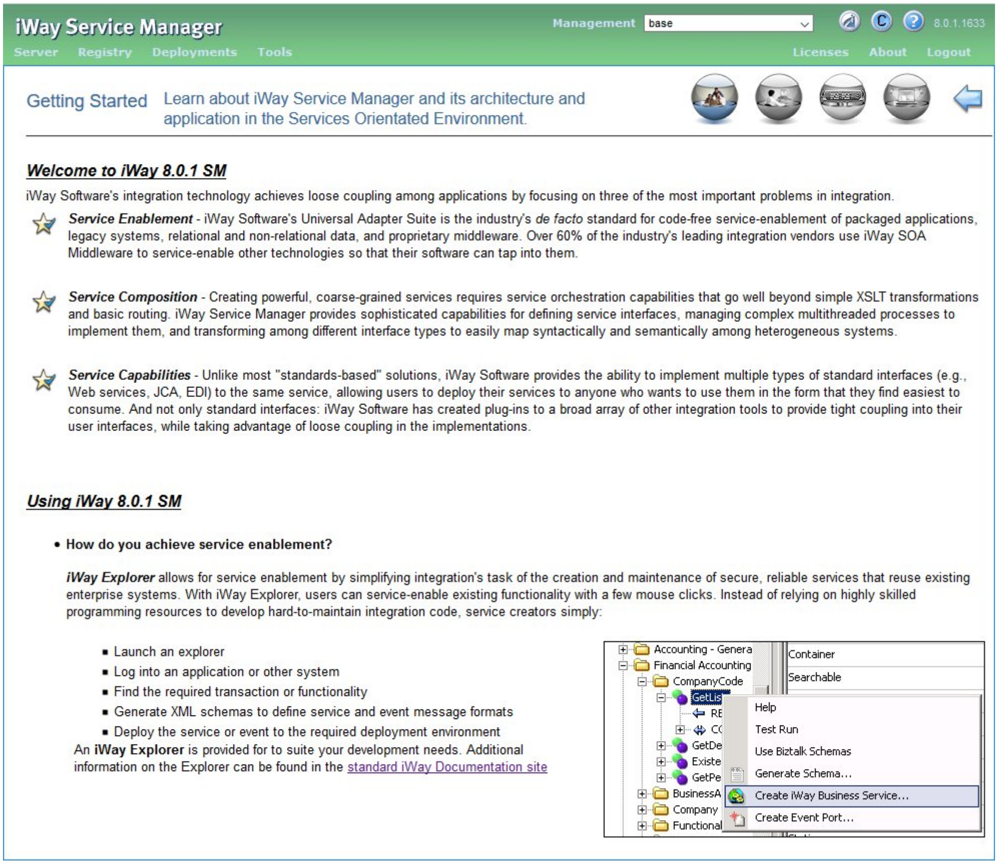
This Welcome window provides the following topics that you can explore:

- **Getting Started** - Learn about iWay Service Manager’s architecture and application in the Service-Oriented environment.
- **Tutorials** - Learn how to be productive using iWay Service Manager by completing end-to-end tutorials that will guide you along the way.
- **Info Center** - Provides access to iWay Service Manager’s documentation.
- **What’s New** - Discover what's new and what's improved in this version of iWay Service Manager.

2. Click a corresponding topic icon in the window to access that topic’s page.

Getting Started

The Getting Started page provides an overview of iWay Service Manager and describes the fundamental concepts.



Getting Started Learn about iWay Service Manager and its architecture and application in the Services Orientated Environment.

Welcome to iWay 8.0.1 SM

iWay Software's integration technology achieves loose coupling among applications by focusing on three of the most important problems in integration.

☆ ***Service Enablement*** - iWay Software's Universal Adapter Suite is the industry's *de facto* standard for code-free service-enablement of packaged applications, legacy systems, relational and non-relational data, and proprietary middleware. Over 60% of the industry's leading integration vendors use iWay SOA Middleware to service-enable other technologies so that their software can tap into them.

☆ ***Service Composition*** - Creating powerful, coarse-grained services requires service orchestration capabilities that go well beyond simple XSLT transformations and basic routing. iWay Service Manager provides sophisticated capabilities for defining service interfaces, managing complex multithreaded processes to implement them, and transforming among different interface types to easily map syntactically and semantically among heterogeneous systems.

☆ ***Service Capabilities*** - Unlike most "standards-based" solutions, iWay Software provides the ability to implement multiple types of standard interfaces (e.g., Web services, JCA, EDI) to the same service, allowing users to deploy their services to anyone who wants to use them in the form that they find easiest to consume. And not only standard interfaces: iWay Software has created plug-ins to a broad array of other integration tools to provide tight coupling into their user interfaces, while taking advantage of loose coupling in the implementations.

Using iWay 8.0.1 SM

- **How do you achieve service enablement?**

  iWay Explorer allows for service enablement by simplifying integration's task of the creation and maintenance of secure, reliable services that reuse existing enterprise systems. With iWay Explorer, users can service-enable existing functionality with a few mouse clicks. Instead of relying on highly skilled programming resources to develop hard-to-maintain integration code, service creators simply:

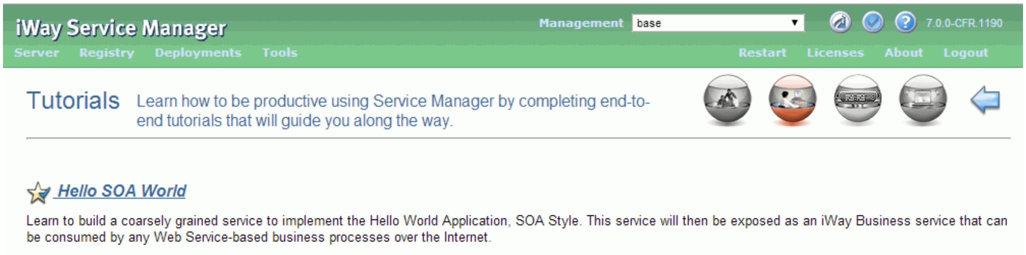
  - Launch an explorer
  - Log into an application or other system
  - Find the required transaction or functionality
  - Generate XML schemas to define service and event message formats
  - Deploy the service or event to the required deployment environment

  An **iWay Explorer** is provided for to suite your development needs. Additional information on the Explorer can be found in the standard iWay Documentation site

To access the remaining topics that are available in the iWay Service Manager Welcome window, click the corresponding icon at the top of the page. You can also click the left blue arrow to return to the Welcome window.

Tutorials

The Tutorials page contains a link to a tutorial called *Hello SOA World*, which guides you through the process of building a service to implement the Hello World application using the Service-Oriented Architecture (SOA) provided by iWay Service Manager. This service will be exposed as an iWay Business Service that can be consumed by any web service-based business process over the Internet.

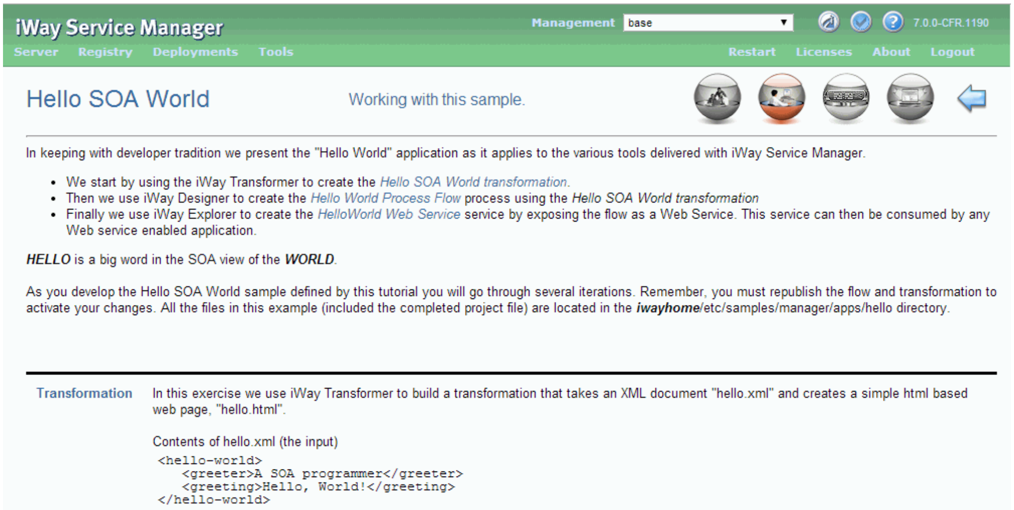


Tutorials Learn how to be productive using Service Manager by completing end-to-end tutorials that will guide you along the way.

Hello SOA World

Learn to build a coarsely grained service to implement the Hello World Application, SOA Style. This service will then be exposed as an iWay Business service that can be consumed by any Web Service-based business processes over the Internet.

To access this tutorial, click the *Hello SOA World* link. The following page opens.



Hello SOA World Working with this sample.

In keeping with developer tradition we present the "Hello World" application as it applies to the various tools delivered with iWay Service Manager.

- We start by using the iWay Transformer to create the *Hello SOA World transformation*.
- Then we use iWay Designer to create the *Hello World Process Flow* process using the *Hello SOA World transformation*
- Finally we use iWay Explorer to create the *HelloWorld Web Service* service by exposing the flow as a Web Service. This service can then be consumed by any Web service enabled application.

HELLO is a big word in the SOA view of the ***WORLD***.

As you develop the Hello SOA World sample defined by this tutorial you will go through several iterations. Remember, you must republish the flow and transformation to activate your changes. All the files in this example (included the completed project file) are located in the ***iwayhome***/etc/samples/manager/apps/hello directory.

Transformation In this exercise we use iWay Transformer to build a transformation that takes an XML document "hello.xml" and creates a simple html based web page, "hello.html".

Contents of hello.xml (the input)

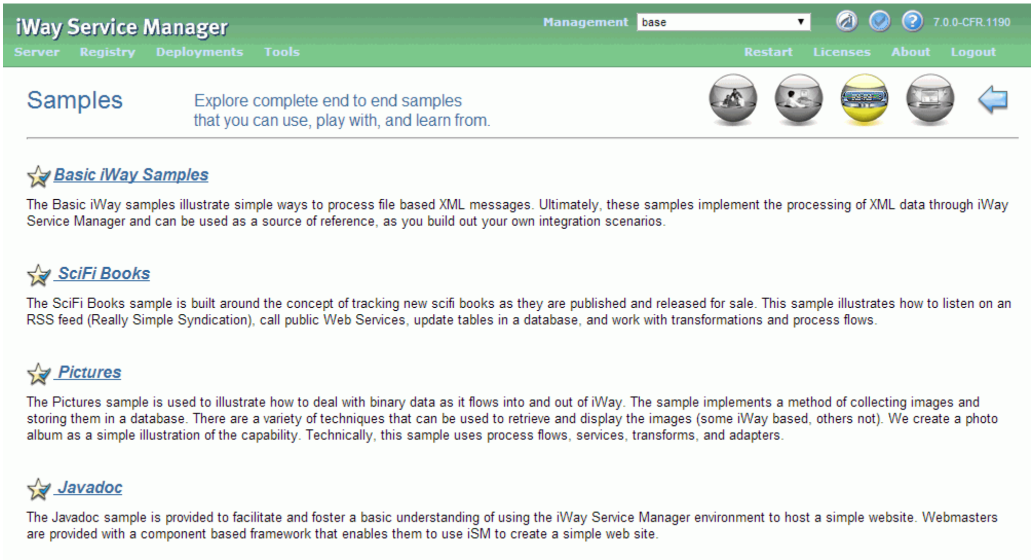
```
<hello-world>
   <greeter>A SOA programmer</greeter>
   <greeting>Hello, World!</greeting>
</hello-world>
```

All the necessary instructions required to build the required transformation, process flow, and service are provided.

To access the remaining topics that are available in the iWay Service Manager Welcome window, click the corresponding icon at the top of the page. You can also click the left blue arrow to return to the Welcome window.

Samples

The Samples page contains links to a variety of samples that are provided with iWay Service Manager. These samples are complete end-to-end solutions that use pre-defined iWay components available in the iWay Service Manager Administration Console.



Samples

Explore complete end to end samples that you can use, play with, and learn from.

Basic iWay Samples

The Basic iWay samples illustrate simple ways to process file based XML messages. Ultimately, these samples implement the processing of XML data through iWay Service Manager and can be used as a source of reference, as you build out your own integration scenarios.

SciFi Books

The SciFi Books sample is built around the concept of tracking new scifi books as they are published and released for sale. This sample illustrates how to listen on an RSS feed (Really Simple Syndication), call public Web Services, update tables in a database, and work with transformations and process flows.

Pictures

The Pictures sample is used to illustrate how to deal with binary data as it flows into and out of iWay. The sample implements a method of collecting images and storing them in a database. There are a variety of techniques that can be used to retrieve and display the images (some iWay based, others not). We create a photo album as a simple illustration of the capability. Technically, this sample uses process flows, services, transforms, and adapters.

Javadoc

The Javadoc sample is provided to facilitate and foster a basic understanding of using the iWay Service Manager environment to host a simple website. Webmasters are provided with a component based framework that enables them to use iSM to create a simple web site.

The following samples are available:

- **Basic iWay Samples** - Illustrate simple ways to process file-based XML messages. Ultimately, these samples implement the processing of XML data through iWay Service Manager and can be used as a reference, as you build your own integration scenarios.

- **SciFi Books** - Illustrates how to track new science fiction books as they are published and released for sale. This sample describes how to listen on an RSS feed (Really Simple Syndication), call public web services, update tables in a database, and work with transformations and process flows.

- **Pictures** - Illustrates a simple way to load and retrieve a set of images into and from a database. This sample describes handling binary data through iWay Service Manager and can be a source of reference as you build your own flows.

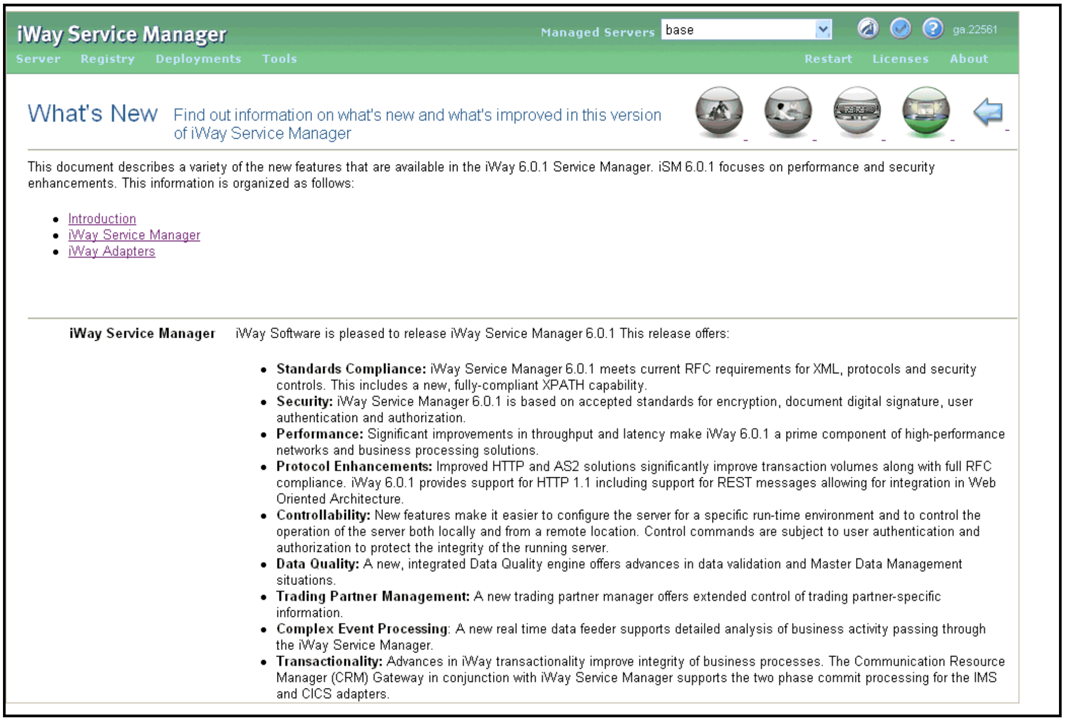
- **Javadoc** - Illustrates how to use iWay Service Manager to host a simple website. Webmasters are provided with a component-based framework that enables them to use iWay Service Manager to create a simple website.

To access a sample, click the name of the sample you want to explore.

To access the remaining topics that are available in the iWay Service Manager Welcome window, click the corresponding icon at the top of the page. You can also click the left blue arrow to return to the Welcome window.

What's New

The What's New page lists and describes new features in this version of iWay Service Manager.



What's New Find out information on what's new and what's improved in this version of iWay Service Manager

This document describes a variety of the new features that are available in the iWay 6.0.1 Service Manager. iSM 6.0.1 focuses on performance and security enhancements. This information is organized as follows:

- Introduction
- iWay Service Manager
- iWay Adapters

iWay Service Manager iWay Software is pleased to release iWay Service Manager 6.0.1 This release offers:

- **Standards Compliance:** iWay Service Manager 6.0.1 meets current RFC requirements for XML, protocols and security controls. This includes a new, fully-compliant XPATH capability.
- **Security:** iWay Service Manager 6.0.1 is based on accepted standards for encryption, document digital signature, user authentication and authorization.
- **Performance:** Significant improvements in throughput and latency make iWay 6.0.1 a prime component of high-performance networks and business processing solutions.
- **Protocol Enhancements:** Improved HTTP and AS2 solutions significantly improve transaction volumes along with full RFC compliance. iWay 6.0.1 provides support for HTTP 1.1 including support for REST messages allowing for integration in Web Oriented Architecture.
- **Controllability:** New features make it easier to configure the server for a specific run-time environment and to control the operation of the server both locally and from a remote location. Control commands are subject to user authentication and authorization to protect the integrity of the running server.
- **Data Quality:** A new, integrated Data Quality engine offers advances in data validation and Master Data Management situations.
- **Trading Partner Management:** A new trading partner manager offers extended control of trading partner-specific information.
- **Complex Event Processing**: A new real time data feeder supports detailed analysis of business activity passing through the iWay Service Manager.
- **Transactionality:** Advances in iWay transactionality improve integrity of business processes. The Communication Resource Manager (CRM) Gateway in conjunction with iWay Service Manager supports the two phase commit processing for the IMS and CICS adapters.

The information is organized as follows:

❑ Introduction

❑ iWay Service Manager

❑ iWay Adapters

To access the remaining topics that are available in the iWay Service Manager Welcome window, click the corresponding icon at the top of the page. You can also click the left blue arrow to return to the Welcome window.

Internationalization

iWay provides a simple framework to access localized text and to label messages.

Internationalization features rely on encoding. For more information, see *Encoding* on page 49.

From the console, you can:

❑ Select the language to be used to display console information.

The Language field in the console home window shows the language used by the console. The default language is English. For instructions on how to change the language, see *How to Select the Language Displayed by the Console* on page 49.

❑ Set the default encoding that iWay Service Manager uses when the encoding scheme is not set on an incoming message.

For instructions, see *How to Set the Default Encoding on a Listener* on page 51.

Procedure: How to Select the Language Displayed by the Console

To select a language:

1. In the General Properties pane, select a language from the Language drop-down list, then click *Save*.
2. Stop and restart the server to apply the change.

Encoding

A message consists of a sequence of characters. A character itself is an abstract notion. A character is defined by the assignment of a group of bits to a glyph, or to an instance of a character that can be displayed. Encoding refers to the sequence of bits assigned to represent related characters.

There are eight bits in a byte and a limited number of characters that these bits can represent. As a result, the same sequence of bits often is assigned to multiple characters. The bits do not refer to the character as unique among other possible characters, but rather to a specific character within a limited group of characters, for example, the letters in a local alphabet such as English, French, or Japanese.

iWay Service Manager must recognize specific characters in a message. Therefore, it is important to identify the exact sequence or group of characters to which the bits belong. Only with this information can iWay Service Manager correctly interpret and process the message.

iWay Service Manager supports all encoding schemes normally used.

Unicode

The document character set for XML and HTML 4.0 is Unicode (also known as ISO-10646). HTML browsers and XML processors use Unicode internally, but documents are not required to be transmitted in Unicode. Provided that the client and server agree on the encoding scheme, the browser or processor can use any encoding that can be converted to Unicode. The character encoding scheme of any XML or (X)HTML document must be clearly labeled. With this information, clients can easily map these encoding schemes to Unicode.

Working With XML Documents

Because XML documents can originate from sources using many languages, the encoding scheme of a specific document is, as a standard, included in the document. Encoding schemes for XML documents are expressed by names assigned by the Internet Assigned Naming Authority (IANA). iWay Service Manager recognizes this encoding declaration and respects it for analysis and handling of the message.

The responsibility for declaring the correct encoding scheme belongs to the originator of the document. An XML message without a specific encoding declaration is given a default encoding scheme by examining the first few characters of the message. The usual default assignment is ASCII or EBCDIC.

Specifying the wrong encoding scheme for a message is a common source of problems and usually results in the inability of iWay Service Manager to parse the message, thus generating an error. For example, it is a common mistake to assign the encoding scheme UTF-8 to every message under the assumption that this is the "cover all cases" scheme.

In reality, UTF-8 is a variable-bit sequence that is very specific; some characters (ASCII 127 and lower) map correctly. However, other characters (above 127) consist of bit patterns that may not be valid UTF-8 encoding. Erroneous use of UTF-8 often results in parsing errors.

Working With Non-XML Documents

A non-XML document does not carry its encoding scheme in a manner that iWay Service Manager can recognize. Such a document may be processed into XML by preparser exits. In this case, iWay Service Manager must recognize which encoding scheme to apply to the message. The listener configuration must specify the encoding scheme to use.

From the console, you can set the default encoding that iWay Service Manager uses when it cannot determine the encoding scheme from an incoming message. For instructions, see *How to Set the Default Encoding on a Listener* on page 51.

Although the engine is optimized for handling XML documents, including non-XML that passes through preparsers to create XML, you can pass non-XML through the engine stages.

A non-XML message is referred to as a *flat* document that, depending on the message, stores the message as a byte array, a string, or an attachment array. A flat document does not pass through the preparser and reviewer exits but is passed through to the business exits.

A common use for a flat document is simple protocol conversion, in which a message is retrieved on one protocol and emitted on another. If no transformation or processing is required, a performance benefit can often be obtained through the elimination of the XML conversion and parsing. Protocol emitters can emit messages from both XML and flat documents.

An incoming message can be established as a flat document by setting the appropriate listener property, so that all documents arriving on that listener are treated as flat. An exit can also store flat information in the document, in which case the document is marked as flat. Another exit can return the document to an XML state by storing an XML tree.

Java Encoding Schemes

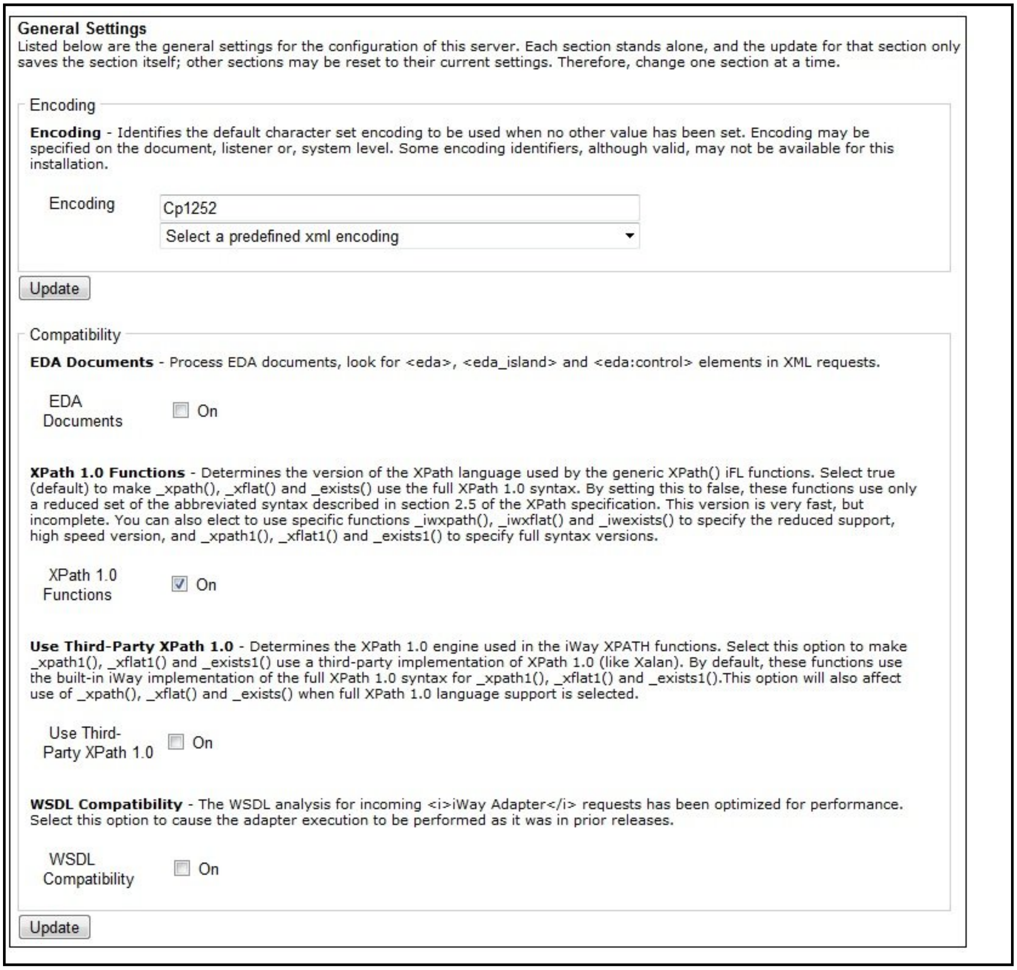
iWay Service Manager processes messages using the Java language and uses the appropriate Java encoding scheme to convert the sequence of bits into usable information. For this reason, iWay Service Manager converts the IANA names to their appropriate Java encoding equivalents. There is a one-to-one mapping from IANA names to Java names; however, there is no mapping in the other direction. In addition, Java names can vary by platform and locale. Therefore, a listener configuration must include the Java encoding name.

Procedure: How to Set the Default Encoding on a Listener

To set the default encoding on a listener:

1. From the main menu, choose *Server*, then *General Settings*.

The General Settings pane opens, as shown in the following image.



General Settings

Listed below are the general settings for the configuration of this server. Each section stands alone, and the update for that section only saves the section itself; other sections may be reset to their current settings. Therefore, change one section at a time.

Encoding

Encoding - Identifies the default character set encoding to be used when no other value has been set. Encoding may be specified on the document, listener or, system level. Some encoding identifiers, although valid, may not be available for this installation.

Encoding: Cp1252

Select a predefined xml encoding

Compatibility

EDA Documents - Process EDA documents, look for <eda>, <eda_island> and <eda:control> elements in XML requests.

EDA Documents ☐ On

XPath 1.0 Functions - Determines the version of the XPath language used by the generic XPath() iFL functions. Select true (default) to make _xpath(), _xflat() and _exists() use the full XPath 1.0 syntax. By setting this to false, these functions use only a reduced set of the abbreviated syntax described in section 2.5 of the XPath specification. This version is very fast, but incomplete. You can also elect to use specific functions _iwxpath(), _iwxflat() and _iwexists() to specify the reduced support, high speed version, and _xpath1(), _xflat1() and _exists1() to specify full syntax versions.

XPath 1.0 Functions ☑ On

Use Third-Party XPath 1.0 - Determines the XPath 1.0 engine used in the iWay XPATH functions. Select this option to make _xpath1(), _xflat1() and _exists1() use a third-party implementation of XPath 1.0 (like Xalan). By default, these functions use the built-in iWay implementation of the full XPath 1.0 syntax for _xpath1(), _xflat1() and _exists1().This option will also affect use of _xpath(), _xflat() and _exists() when full XPath 1.0 language support is selected.

Use Third-Party XPath 1.0 ☐ On

WSDL Compatibility - The WSDL analysis for incoming <i>iWay Adapter</i> requests has been optimized for performance. Select this option to cause the adapter execution to be performed as it was in prior releases.

WSDL Compatibility ☐ On

2. In the Encoding field, select the default encoding by choosing the encoding name from the drop-down list or typing it directly.

   The default value is the platform encoding scheme used to read and write characters in the native file system and depends on the platform on which iWay Service Manager runs.

   The specified encoding scheme must be available for use by iWay Service Manager. Encoding schemes are provided to Java in the I18N.jar file. You must obtain the appropriate I18N.jar for your locale and platform and load it into the iWay lib directory. The JAR file can be obtained from:

   `www.javasoft.com`

3. Click *Update*.

Displaying License Information

A license file determines the iWay products you are authorized to run. The license file for your installation resides in the iWay7 home directory.

To display information about your license file, select the *Licenses* option from the upper right of the console. A window opens, showing the name of your license file and licensed feature codes. If applicable, the window displays the names of unregistered license files and provides error messages, such as “license file invalid”.

Registering iWay Service Manager

After installation, you must register the product to obtain a permanent license. You can provide configuration information in the configuration webpages that are provided. These pages run as part of iWay Service Manager.

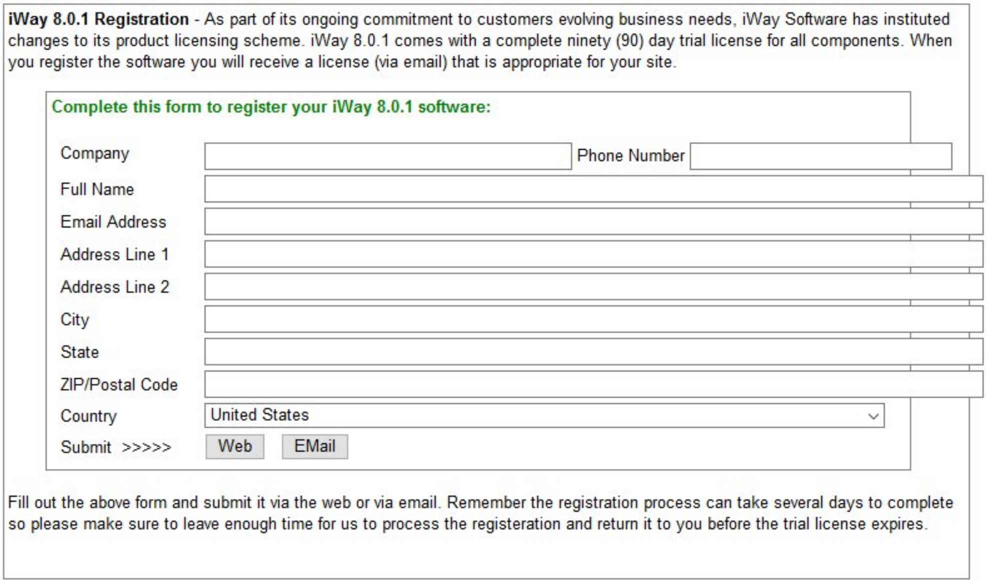
iWay Service Manager includes a ninety (90) day trial license for all components.

Procedure: How to Register iWay

To register iWay:

1. If it is not already started, start iWay Service Manager.
2. Click *Licenses* in the upper right of the pane.
   The Licensed Features pane opens.
3. Click *register the software*.

The iWay Registration pane opens, as shown in the following image.



iWay 8.0.1 Registration - As part of its ongoing commitment to customers evolving business needs, iWay Software has instituted changes to its product licensing scheme. iWay 8.0.1 comes with a complete ninety (90) day trial license for all components. When you register the software you will receive a license (via email) that is appropriate for your site.

Complete this form to register your iWay 8.0.1 software:

Company		Phone Number	
Full Name			
Email Address			
Address Line 1			
Address Line 2			
City			
State			
ZIP/Postal Code			
Country	United States		
Submit >>>>>	Web	EMail	

Fill out the above form and submit it via the web or via email. Remember the registration process can take several days to complete so please make sure to leave enough time for us to process the registeration and return it to you before the trial license expires.

4. Supply the requested information.
5. Click *Web* or *Email* to submit the information.

Chapter 3

Installation Verification Procedure

The Installation Verification Procedure (IVP) enables you to ensure that iWay Service Manager (iSM) is installed and configured correctly.

To verify iSM installation and configuration:

1. Start the default configuration, which is called *base*, and log on to the iWay Service Manager Administration Console.
2. Build and deploy the file1 channel, which is included as a sample.
3. Test iWay Service Manager by using the channel to move an XML file from an input directory to an output directory.

The file1 channel sample illustrates the simplest form of a channel that picks up an XML document from the file system and moves it to another location in the file system.

A variety of samples are included with iWay Service Manager to help you achieve a basic understanding of the iWay Service Manager environment and the concepts that are used. For more information on samples, see *Introducing iWay Service Manager* on page 13.

For more information on installing iWay Service Manager, see the *iWay Installation and Configuration Guide*.

In this chapter:

Accessing the iWay Service Manager Administration Console

The iWay Service Manager Administration Console is a browser-based interface that is used to design, configure, and deploy components.

Procedure: How to Access the iWay Service Manager Administration Console

To access the iWay Service Manager Administration Console:

1. Ensure that iWay Service Manager is started.

   For more information on starting and stopping iWay Service Manager, see *Operations and Monitoring* on page 389.

2. Open a web browser and point to the following URL:

   `http://host:port`

   where:

   host

   Is the host machine where iWay Service Manager is installed. The default value is *localhost*.

   port

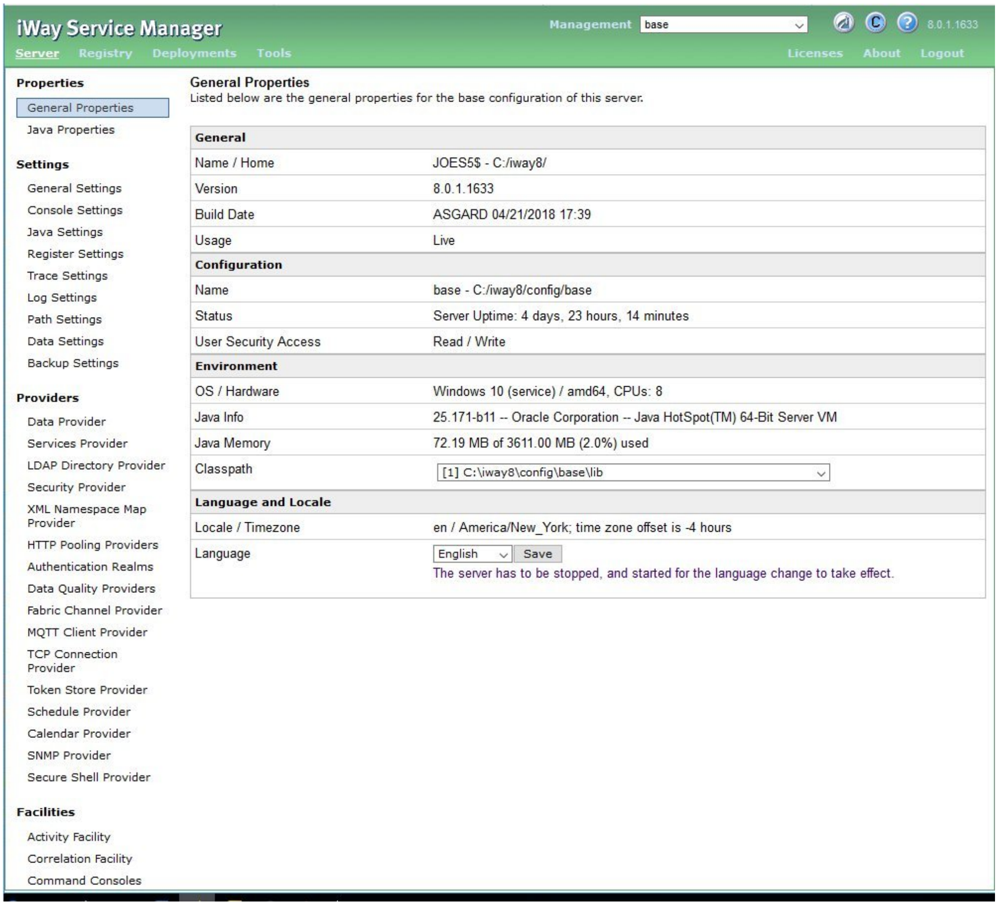
   Is the port where iWay Service Manager is listening. The default port is *9999*.

   Alternatively, on Windows, you can select *Start*, *Programs*, *iWay 7 Service Manager*, and then *Console*.

   A logon dialog box opens.

3. Type *iway* for both the user name and password, and then, click *OK*.

   The iWay Service Manager Administration Console opens, showing the General Properties section in the central pane.



General Properties

Listed below are the general properties for the base configuration of this server.

General	
Name / Home	JOES5$ - C:/iway8/
Version	8.0.1.1633
Build Date	ASGARD 04/21/2018 17:39
Usage	Live
Configuration	
Name	base - C:/iway8/config/base
Status	Server Uptime: 4 days, 23 hours, 14 minutes
User Security Access	Read / Write
Environment	
OS / Hardware	Windows 10 (service) / amd64, CPUs: 8
Java Info	25.171-b11 -- Oracle Corporation -- Java HotSpot(TM) 64-Bit Server VM
Java Memory	72.19 MB of 3611.00 MB (2.0%) used
Classpath	[1] C:\iway8\config\base\lib
Language and Locale	
Locale / Timezone	en / America/New_York; time zone offset is -4 hours
Language	English Save The server has to be stopped, and started for the language change to take effect.

Building and Deploying the File1 Channel

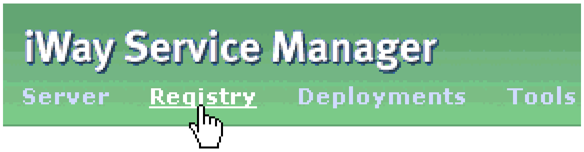
The file1 channel is included as a sample in the registry. This channel has an inlet that includes the *file1* sample listener. The route, called *move*, moves the input stream to the output stream. Finally, the outlet, called *default.outlet* defines an empty outlet. An outlet that does not contain an emitter is considered to be a default outlet. With a default outlet, the output is routed back to the default location specified by the listener.

The following procedure describes how to build and deploy the file1 channel for use in the installation verification process.

Procedure: How to Build and Deploy the File1 Channel

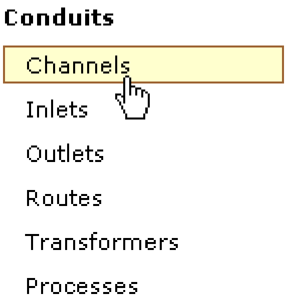
To build and deploy the file1 channel:

1. Click *Registry* in the menu bar, which is located in the top pane.



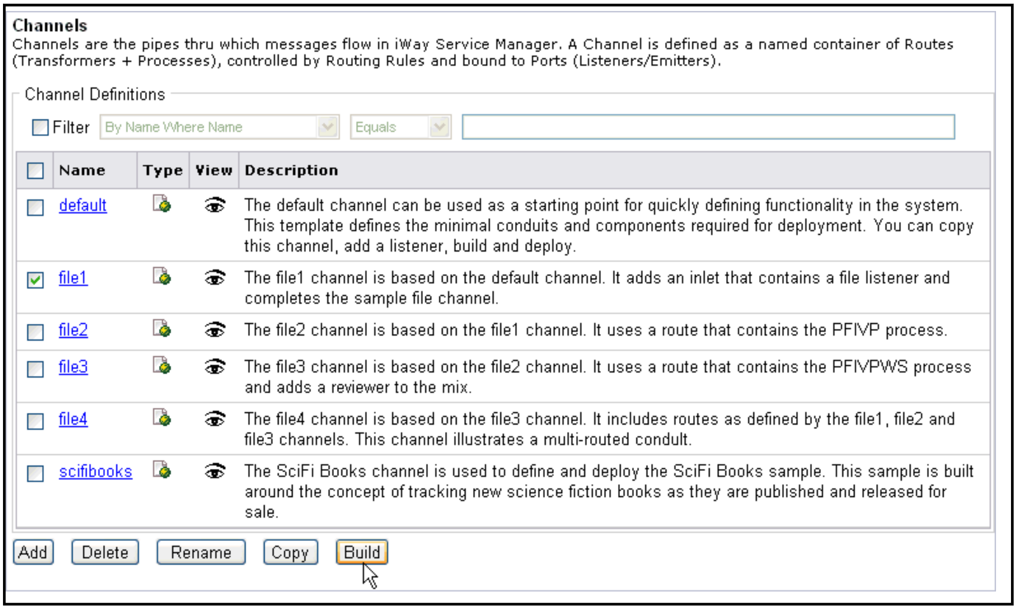
The Registry - Repository pane opens, showing links to various types of conduits and components you can configure.

2. Click *Channels*.



The Channels pane opens.

3. Select the check box next to the file1 channel and click *Build*.



Channels

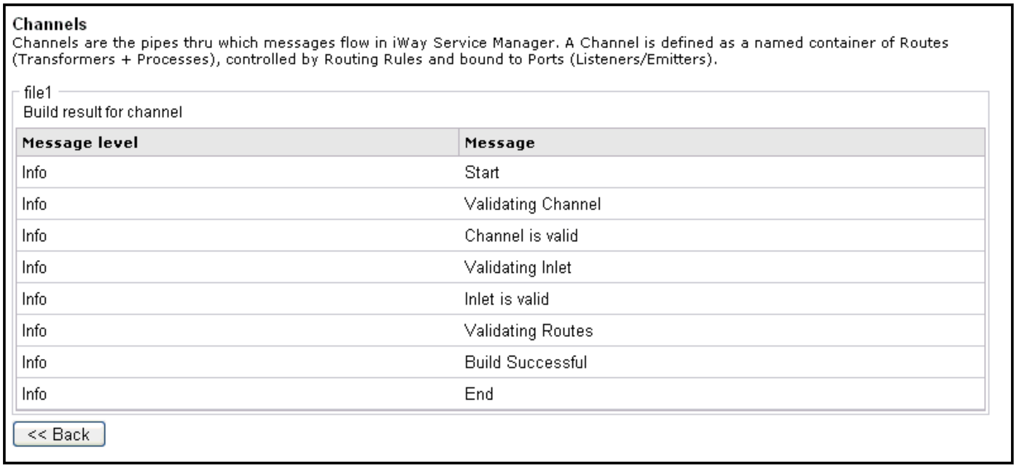
Channels are the pipes thru which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

Channel Definitions

□ Filter By Name Where Name | Equals

□	Name	Type	View	Description
□	default			The default channel can be used as a starting point for quickly defining functionality in the system. This template defines the minimal conduits and components required for deployment. You can copy this channel, add a listener, build and deploy.
☑	file1			The file1 channel is based on the default channel. It adds an inlet that contains a file listener and completes the sample file channel.
□	file2			The file2 channel is based on the file1 channel. It uses a route that contains the PFIVP process.
□	file3			The file3 channel is based on the file2 channel. It uses a route that contains the PFIVPWS process and adds a reviewer to the mix.
□	file4			The file4 channel is based on the file3 channel. It includes routes as defined by the file1, file2 and file3 channels. This channel illustrates a multi-routed conduit.
□	scifibooks			The SciFi Books channel is used to define and deploy the SciFi Books sample. This sample is built around the concept of tracking new science fiction books as they are published and released for sale.

The build result pane for the file1 channel opens.



Channels

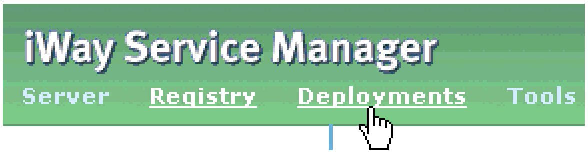
Channels are the pipes thru which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

file1

Build result for channel

Message level	Message
Info	Start
Info	Validating Channel
Info	Channel is valid
Info	Validating Inlet
Info	Inlet is valid
Info	Validating Routes
Info	Build Successful
Info	End

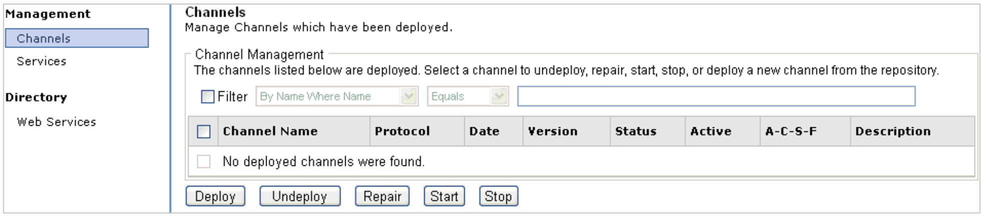
4. Click *Deployments* in the menu bar.



The Deployments pane opens, showing links to Channels, Services, and Web Services, in the left pane.

5. Click *Channels* in the left pane.

The Channel Management pane opens, as shown in the following image.



Channels

Manage Channels which have been deployed.

Channel Management

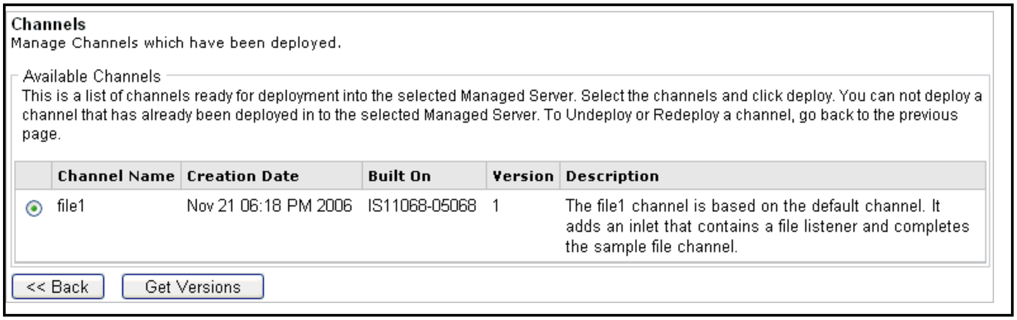
The channels listed below are deployed. Select a channel to undeploy, repair, start, stop, or deploy a new channel from the repository.

□	Channel Name	Protocol	Date	Version	Status	Active	A-C-S-F	Description
□	No deployed channels were found.							

Notice that there are no deployed channels available in the table.

6. Click *Deploy*.

The Available Channels pane opens.



Channels

Manage Channels which have been deployed.

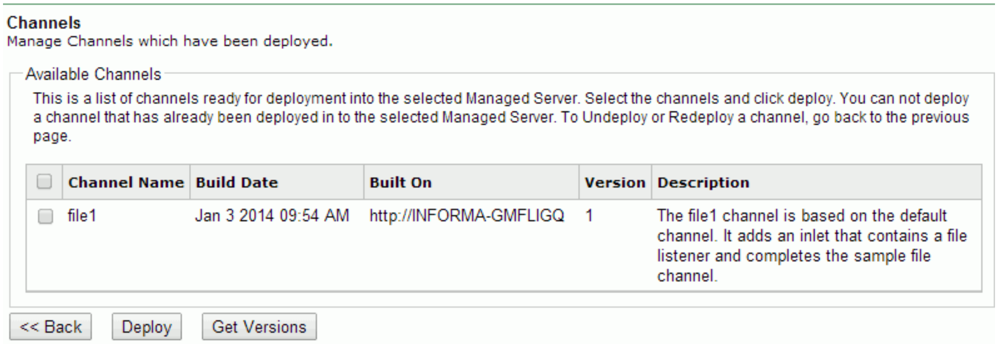
Available Channels

This is a list of channels ready for deployment into the selected Managed Server. Select the channels and click deploy. You can not deploy a channel that has already been deployed in to the selected Managed Server. To Undeploy or Redeploy a channel, go back to the previous page.

	Channel Name	Creation Date	Built On	Version	Description
●	file1	Nov 21 06:18 PM 2006	IS11068-05068	1	The file1 channel is based on the default channel. It adds an inlet that contains a file listener and completes the sample file channel.

7. Select the file1 channel and click *Get Versions*.

   The Channel Versions pane opens.



Channels

Manage Channels which have been deployed.

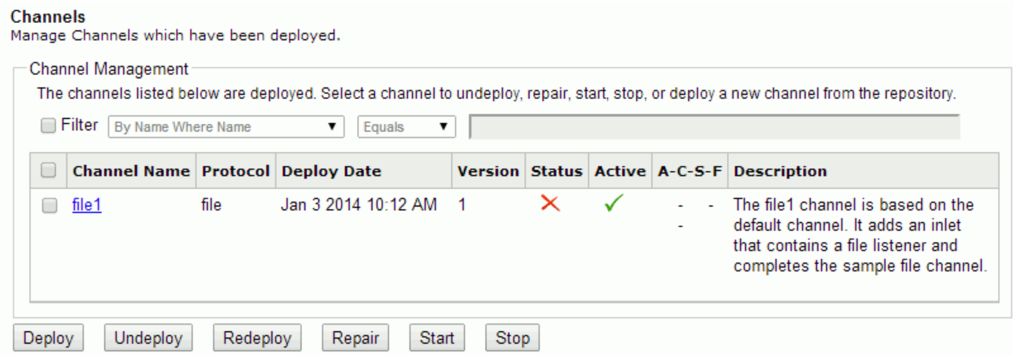
Available Channels

This is a list of channels ready for deployment into the selected Managed Server. Select the channels and click deploy. You can not deploy a channel that has already been deployed in to the selected Managed Server. To Undeploy or Redeploy a channel, go back to the previous page.

□	Channel Name	Build Date	Built On	Version	Description
□	file1	Jan 3 2014 09:54 AM	http://INFORMA-GMFLIGQ	1	The file1 channel is based on the default channel. It adds an inlet that contains a file listener and completes the sample file channel.

8. Select the check box next to the file1 channel and click *Deploy*.

   You are returned to the Channel Management pane. Notice that the file1 channel is now included in the list of deployed channels.



Channels

Manage Channels which have been deployed.

Channel Management

The channels listed below are deployed. Select a channel to undeploy, repair, start, stop, or deploy a new channel from the repository.

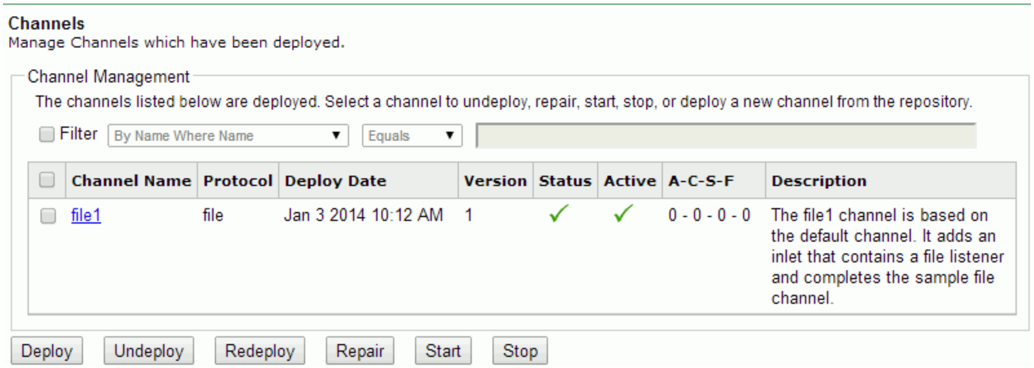
□ Filter: By Name Where Name ▼ Equals ▼

□	Channel Name	Protocol	Deploy Date	Version	Status	Active	A-C-S-F	Description
□	file1	file	Jan 3 2014 10:12 AM	1	✗	✓	- - -	The file1 channel is based on the default channel. It adds an inlet that contains a file listener and completes the sample file channel.

Note: Once a channel is deployed, you must also start it. A deployed channel is not started automatically. Notice the red X in the Status column.

9. Select the check box next to the deployed file1 channel and click *Start*.

   A green checkmark now displays in the Status column, indicating that the deployed channel is started.



Channels

Manage Channels which have been deployed.

Channel Management

The channels listed below are deployed. Select a channel to undeploy, repair, start, stop, or deploy a new channel from the repository.

□ Filter By Name Where Name ▼ Equals ▼

□	Channel Name	Protocol	Deploy Date	Version	Status	Active	A-C-S-F	Description
□	file1	file	Jan 3 2014 10:12 AM	1	✓	✓	0 - 0 - 0 - 0	The file1 channel is based on the default channel. It adds an inlet that contains a file listener and completes the sample file channel.

Testing iWay Service Manager

You can use the file1 channel that was deployed in *How to Build and Deploy the File1 Channel* on page 58 to verify iWay Service Manager.

The file1 listener included in the inlet of the file1 channel picks up the XML document placed in the specified input directory and moves it into a specified output directory.

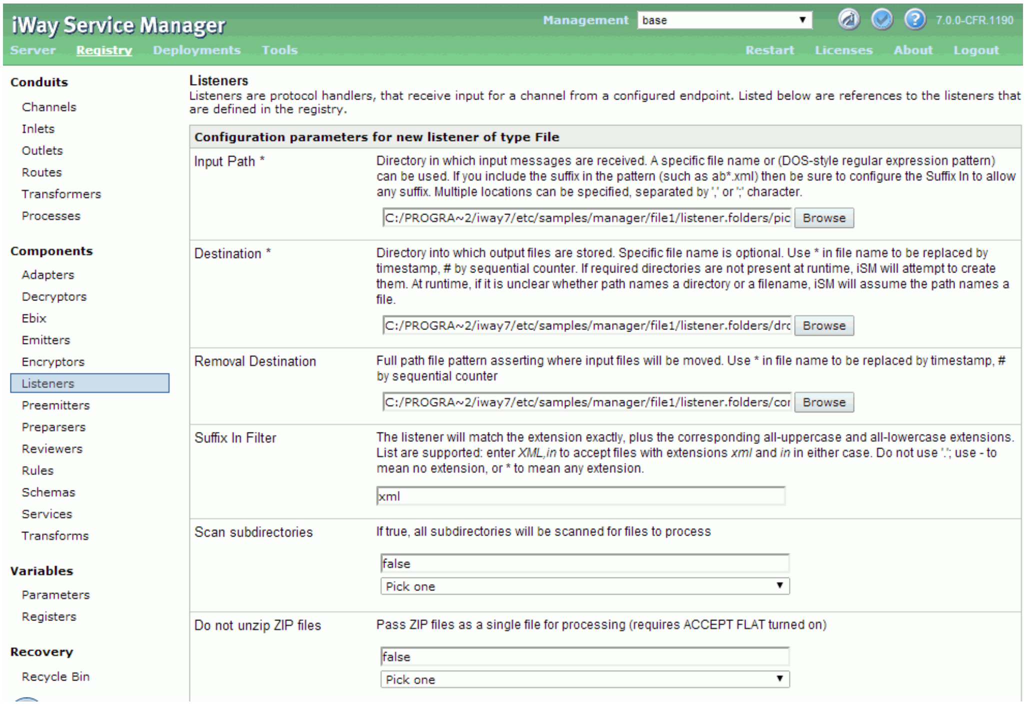
Procedure: How to Test iWay Service Manager Using the File1 Channel

To test iWay Service Manager using the file1 channel:

1. Use an edit utility such as Notepad to create the following XML file:

```
<test>
This is an XML test file.
</test>
```

2. Save the XML file into the directory specified as the input path for the file1 listener.



Listeners

Listeners are protocol handlers, that receive input for a channel from a configured endpoint. Listed below are references to the listeners that are defined in the registry.

Configuration parameters for new listener of type File	
Input Path *	Directory in which input messages are received. A specific file name or (DOS-style regular expression pattern) can be used. If you include the suffix in the pattern (such as ab*.xml) then be sure to configure the Suffix In to allow any suffix. Multiple locations can be specified, separated by ',' or ';' character. C:/PROGRA~2/iway7/etc/samples/manager/file1/listener.folders/pic Browse
Destination *	Directory into which output files are stored. Specific file name is optional. Use * in file name to be replaced by timestamp, # by sequential counter. If required directories are not present at runtime, iSM will attempt to create them. At runtime, if it is unclear whether path names a directory or a filename, iSM will assume the path names a file. C:/PROGRA~2/iway7/etc/samples/manager/file1/listener.folders/drc Browse
Removal Destination	Full path file pattern asserting where input files will be moved. Use * in file name to be replaced by timestamp, # by sequential counter C:/PROGRA~2/iway7/etc/samples/manager/file1/listener.folders/cor Browse
Suffix In Filter	The listener will match the extension exactly, plus the corresponding all-uppercase and all-lowercase extensions. List are supported: enter *XML,in* to accept files with extensions *xml* and *in* in either case. Do not use '.'; use - to mean no extension, or * to mean any extension. xml
Scan subdirectories	If true, all subdirectories will be scanned for files to process false Pick one
Do not unzip ZIP files	Pass ZIP files as a single file for processing (requires ACCEPT FLAT turned on) false Pick one

When viewing the configuration parameters for the file1 listener, the default path that is specified in the Input Path field is:

iwayhome\etc\samples\manager\file1\listener.folders\pickup

where:

iwayhome

Is the name of the directory where iWay Service Manager is installed.

This is the location on your file system where the XML file will be picked up by the file1 listener.

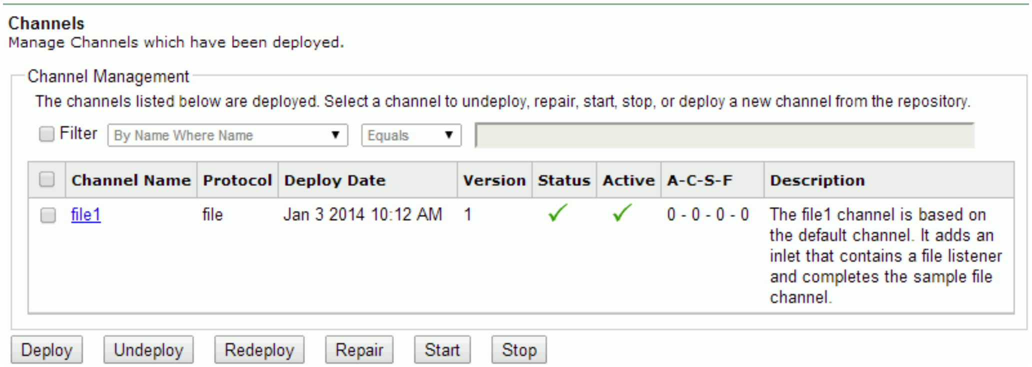
If you correctly installed iWay Service Manager and successfully built and deployed the file1 channel, iWay Service Manager will pick up and then move the XML file into the directory specified as the destination for the file1 listener. The default path is:

iwayhome\etc\samples\manager\file1\listener.folders\dropoff

3. Browse to the destination directory to ensure that the XML file was moved.
Additionally, the monitor pane allows you to verify that the test has completed.

4. Click *Deployments* in the menu bar.

   The Deployments pane opens, showing the Channel Management pane.



Channels
Manage Channels which have been deployed.

Channel Management

The channels listed below are deployed. Select a channel to undeploy, repair, start, stop, or deploy a new channel from the repository.

☐ Filter By Name Where Name ▼ Equals ▼

☐	Channel Name	Protocol	Deploy Date	Version	Status	Active	A-C-S-F	Description
☐	file1	file	Jan 3 2014 10:12 AM	1	✓	✓	0 - 0 - 0 - 0	The file1 channel is based on the default channel. It adds an inlet that contains a file listener and completes the sample file channel.

   Notice the A-C-S-F column, which now shows a 0 below the A (Active), a 1 below the C (Completed), a 1 below the S (Successful), and a 0 below the F (Failed).

   If this is the first XML file you placed in the input directory, and the number for Completed (C) is 1, then your configuration is correct.

You have finished verifying that iWay Service Manager was installed and configured correctly.

Chapter 4

Configuring Basic Properties

This section describes how to configure properties for iWay Service Manager (iSM).

In this chapter:

Configuring Properties

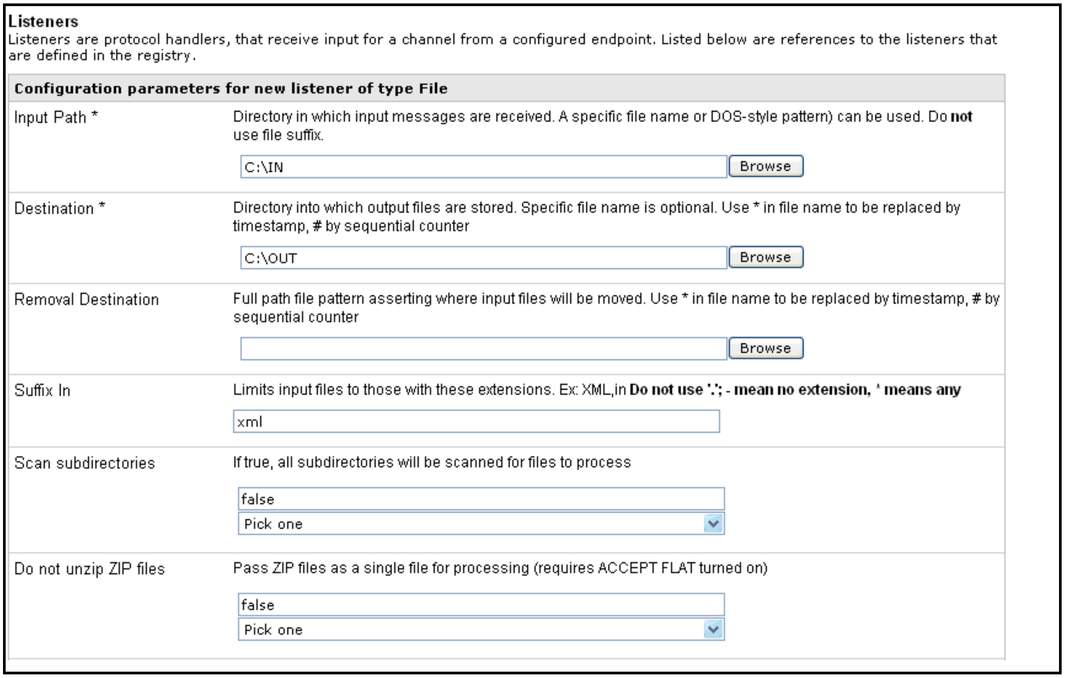
You can provide the properties required to configure iSM in the following ways:

- ❑ From a constant value.
- ❑ From values stored in a file.
- ❑ Using Lightweight Directory Access Protocol (LDAP).
- ❑ Using a document-centric query (XPath) or (JsonPath, JsonPointer).
- ❑ Using a Special Register (SREG).

Configuring Properties as Constant Values

You can provide configuration properties and store them as constant values.

For example, in the Listeners pane, values for the Input Path and Destination for a File listener are displayed.



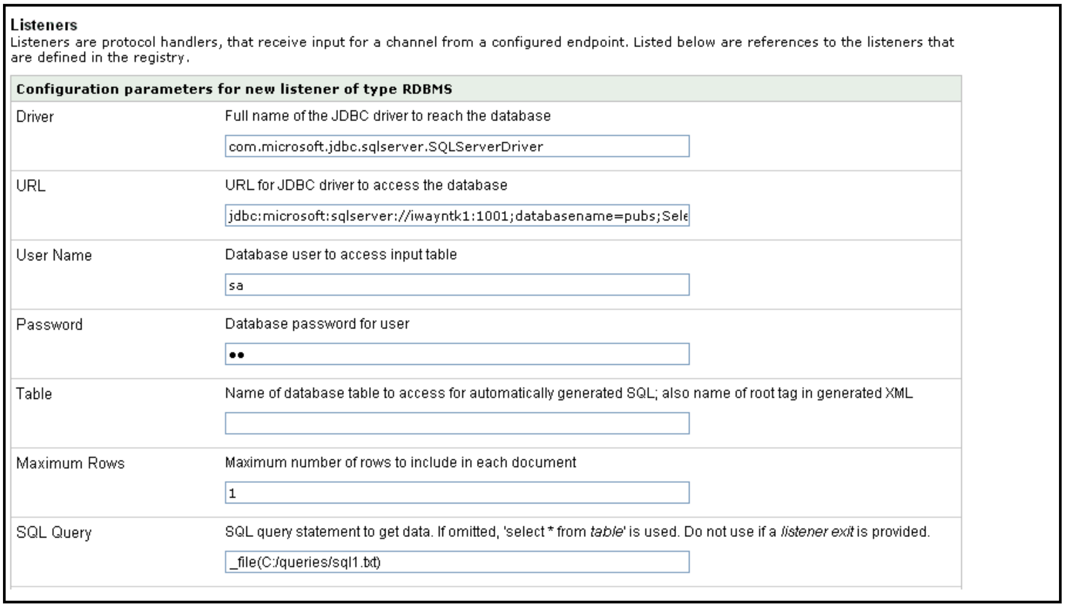
Listeners
Listeners are protocol handlers, that receive input for a channel from a configured endpoint. Listed below are references to the listeners that are defined in the registry.

Configuration parameters for new listener of type File	
Input Path *	Directory in which input messages are received. A specific file name or DOS-style pattern) can be used. Do **not** use file suffix. C:\IN
Destination *	Directory into which output files are stored. Specific file name is optional. Use * in file name to be replaced by timestamp, # by sequential counter C:\OUT
Removal Destination	Full path file pattern asserting where input files will be moved. Use * in file name to be replaced by timestamp, # by sequential counter
Suffix In	Limits input files to those with these extensions. Ex: XML,in **Do not use '.'; - mean no extension, * means any** xml
Scan subdirectories	If true, all subdirectories will be scanned for files to process false Pick one
Do not unzip ZIP files	Pass ZIP files as a single file for processing (requires ACCEPT FLAT turned on) false Pick one

Obtaining Configuration Properties From the File System

A configuration property can be stored in a file on the file system. This method is useful for large, complex queries used with an SQL listener.

The format is `_file(`*`drive:/directory/filename`*`)`. In the RDBMS listener configuration pane, the query resides in the *queries* directory in a file named *sql1.txt*, as shown in the following image.



Listeners

Listeners are protocol handlers, that receive input for a channel from a configured endpoint. Listed below are references to the listeners that are defined in the registry.

Configuration parameters for new listener of type RDBMS

Driver	Full name of the JDBC driver to reach the database	com.microsoft.jdbc.sqlserver.SQLServerDriver
URL	URL for JDBC driver to access the database	jdbc:microsoft:sqlserver://iwayntk1:1001;databasename=pubs;Sele
User Name	Database user to access input table	sa
Password	Database password for user	••
Table	Name of database table to access for automatically generated SQL; also name of root tag in generated XML	
Maximum Rows	Maximum number of rows to include in each document	1
SQL Query	SQL query statement to get data. If omitted, 'select * from *table*' is used. Do not use if a *listener exit* is provided.	_file(C:/queries/sql1.txt)

Obtaining Configuration Properties Using LDAP

LDAP (Lightweight Directory Access Protocol) is a well-established emerging standard for access to corporate directories, such as Microsoft Active Directory and Novell Directory. You can use LDAP to store security information, for example, user IDs and passwords, and configuration properties.

iWay Service Manager (iSM) supports LDAP for looking up parameters to be used in processing exits. In iSM, processing exits include preparsers, preemitters, and services. The LDAP information is resolved at iSM start-up time.

In LDAP, a directory is subdivided into contexts. Within each context, a filter describes a section of the directory from which an attribute is to be obtained. For example, in the iWay Software context, under the filter of <surname='Smith', system='SmithSystem'>, the attribute password would be Smith's password in SmithSystem.

Using LDAP to store configuration properties offers the following benefits:

- There is no duplication of information in the iSM configuration files.
- Any configuration property can be accessed directly from any LDAP-enabled directory.

- As information in the registry changes, the change is automatically propagated to iSM during the next start up, without reconfiguring iSM.

Using LDAP

You can use an LDAP look-up request for most properties in the iSM configuration. To use LDAP, you must define the LDAP directory to iSM. You enter the LDAP provider URL that identifies the path to the LDAP directory and optionally, a root context, for example:

```
ldap://iwaldap:1234/dc=people, do=etc
```

After it is provided, the initial context is used unless it is overridden in an LDAP access request function elsewhere in the configuration.

Some configurations require that you also enter a valid user ID and password on the LDAP directory server. If you request LDAP access and it is not authorized, you cannot start iSM. LDAP servers that are configured to provide anonymous access do not require a user ID or password.

After you receive authorization to use an identified LDAP context, you can specify the value of any property as:

```
_ldap(filter;attribute_to_get[;context])
```

The context is optional, defaulting to the context set in the initial LDAP access specification. Failure to locate the attribute within the context under the filter results in an empty property value.

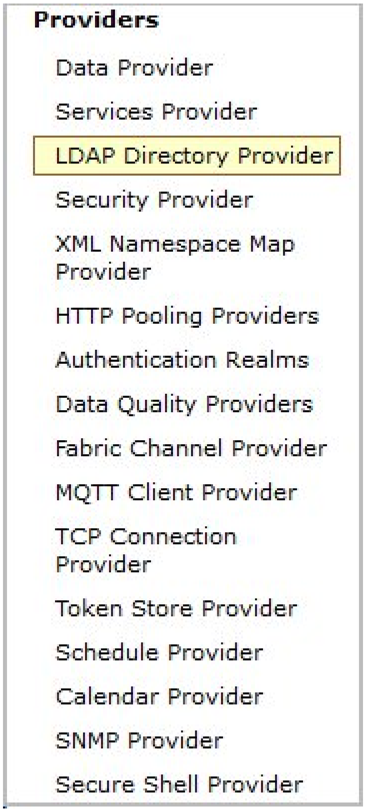
You must configure access to an LDAP server before using LDAP as a means of storing parameters for use by iSM.

Procedure: How to Configure Access to an LDAP Server

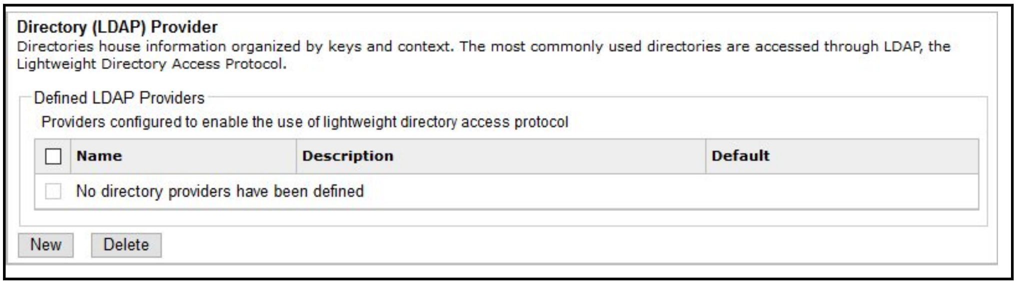
To configure LDAP:

1. In the top pane, click *Server*.

2. From the Providers list in the left pane, click *Directory Provider*.



The Directory Provider pane opens.



Directory (LDAP) Provider

Directories house information organized by keys and context. The most commonly used directories are accessed through LDAP, the Lightweight Directory Access Protocol.

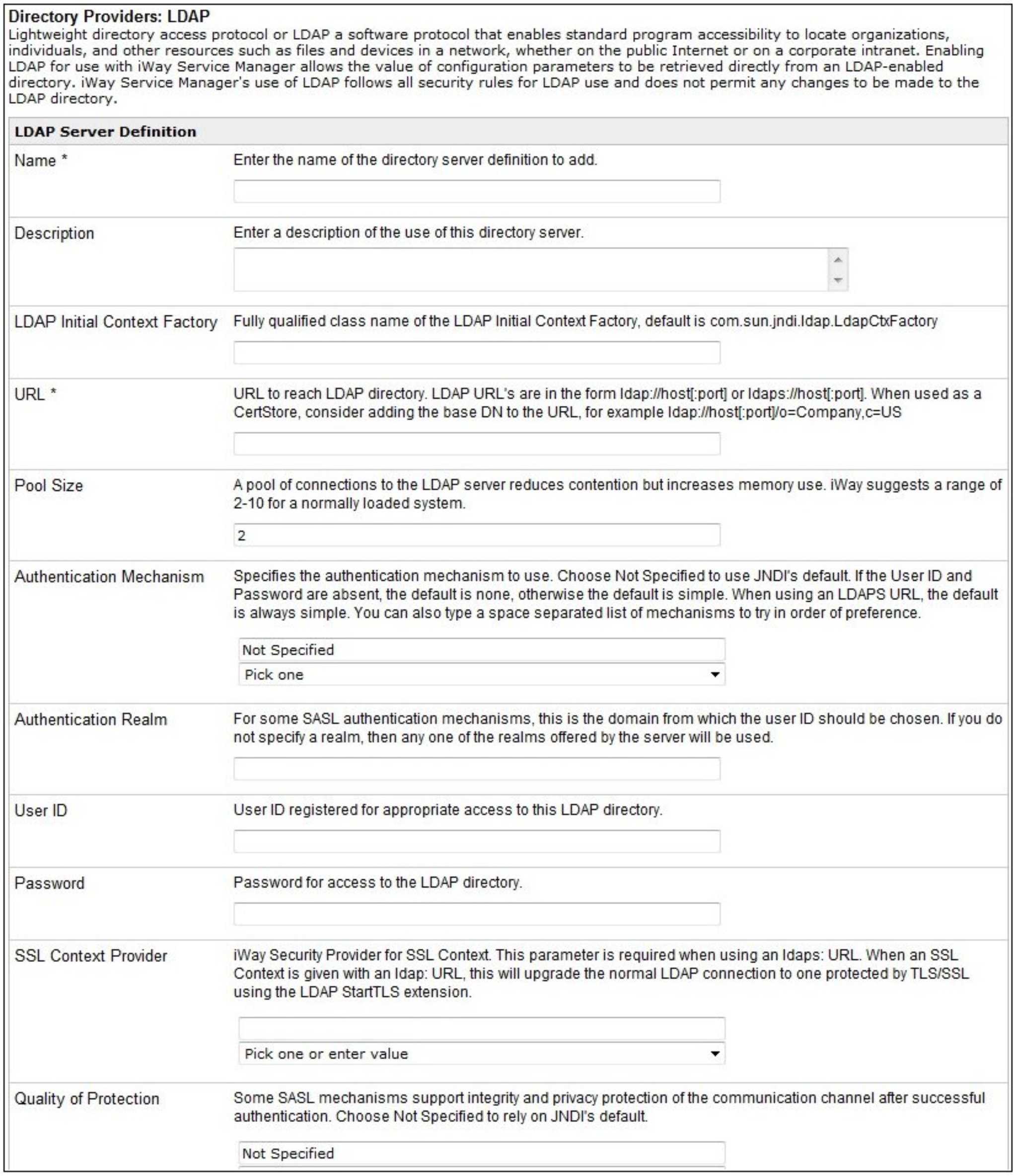
Defined LDAP Providers

Providers configured to enable the use of lightweight directory access protocol

□	Name	Description	Default
□	No directory providers have been defined		

3. Click *New*.

The Directory Providers: LDAP pane opens, as shown in the following image.



Directory Providers: LDAP

Lightweight directory access protocol or LDAP a software protocol that enables standard program accessibility to locate organizations, individuals, and other resources such as files and devices in a network, whether on the public Internet or on a corporate intranet. Enabling LDAP for use with iWay Service Manager allows the value of configuration parameters to be retrieved directly from an LDAP-enabled directory. iWay Service Manager's use of LDAP follows all security rules for LDAP use and does not permit any changes to be made to the LDAP directory.

LDAP Server Definition	
Name *	Enter the name of the directory server definition to add.
Description	Enter a description of the use of this directory server.
LDAP Initial Context Factory	Fully qualified class name of the LDAP Initial Context Factory, default is com.sun.jndi.ldap.LdapCtxFactory
URL *	URL to reach LDAP directory. LDAP URL's are in the form ldap://host[:port] or ldaps://host[:port]. When used as a CertStore, consider adding the base DN to the URL, for example ldap://host[:port]/o=Company,c=US
Pool Size	A pool of connections to the LDAP server reduces contention but increases memory use. iWay suggests a range of 2-10 for a normally loaded system. 2
Authentication Mechanism	Specifies the authentication mechanism to use. Choose Not Specified to use JNDI's default. If the User ID and Password are absent, the default is none, otherwise the default is simple. When using an LDAPS URL, the default is always simple. You can also type a space separated list of mechanisms to try in order of preference. Not Specified Pick one
Authentication Realm	For some SASL authentication mechanisms, this is the domain from which the user ID should be chosen. If you do not specify a realm, then any one of the realms offered by the server will be used.
User ID	User ID registered for appropriate access to this LDAP directory.
Password	Password for access to the LDAP directory.
SSL Context Provider	iWay Security Provider for SSL Context. This parameter is required when using an ldaps: URL. When an SSL Context is given with an ldap: URL, this will upgrade the normal LDAP connection to one protected by TLS/SSL using the LDAP StartTLS extension. Pick one or enter value
Quality of Protection	Some SASL mechanisms support integrity and privacy protection of the communication channel after successful authentication. Choose Not Specified to rely on JNDI's default. Not Specified

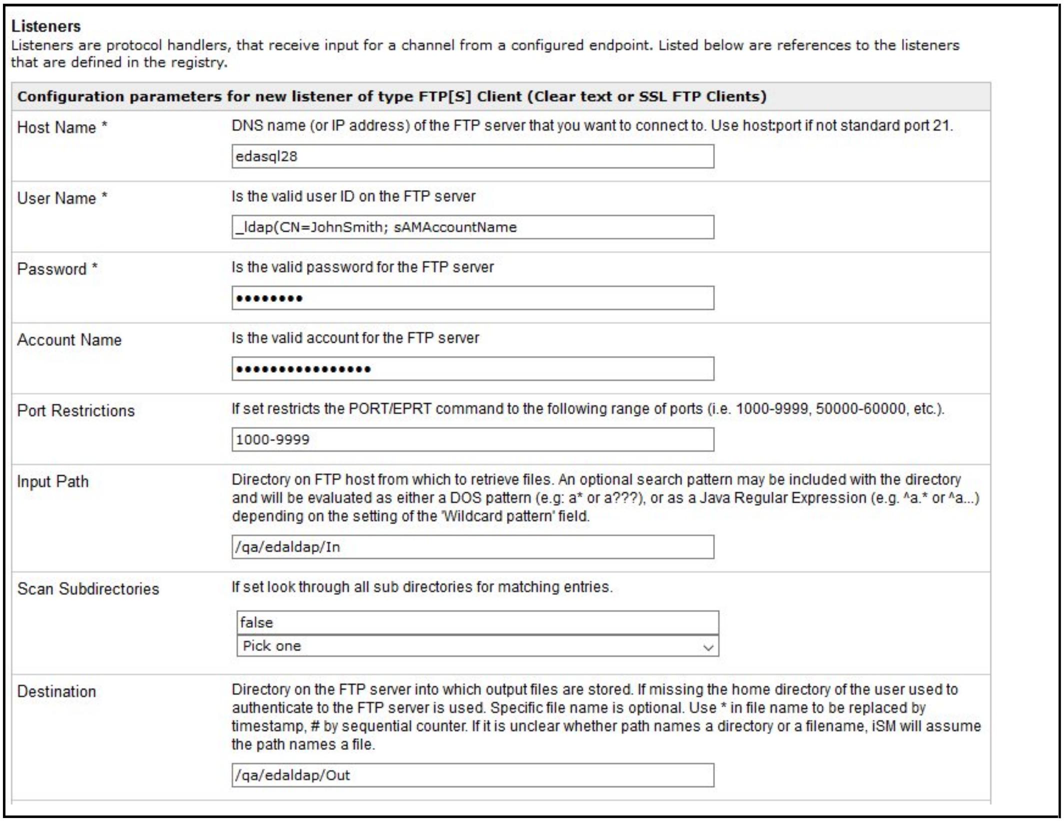
4. Type the property values that are specific to your LDAP server.
5. Click *Test* to verify the provided values, and correct any mistakes if there is a Test Result Failure shown at the top of the pane.

6. When the test is successful, scroll to the bottom of the pane and click *Add*.

Example: Configuring an FTP Listener Using LDAP

Any property can be retrieved using LDAP. The following image shows a sample configuration of an FTP listener where the user name and account name are retrieved from LDAP.

```
_ldap(CN=John Smith; sAMAccountName)
```



Listeners

Listeners are protocol handlers, that receive input for a channel from a configured endpoint. Listed below are references to the listeners that are defined in the registry.

Configuration parameters for new listener of type FTP[S] Client (Clear text or SSL FTP Clients)

Host Name *	DNS name (or IP address) of the FTP server that you want to connect to. Use host:port if not standard port 21.	edasql28
User Name *	Is the valid user ID on the FTP server	_ldap(CN=JohnSmith; sAMAccountName
Password *	Is the valid password for the FTP server	••••••••
Account Name	Is the valid account for the FTP server	••••••••••••••••
Port Restrictions	If set restricts the PORT/EPRT command to the following range of ports (i.e. 1000-9999, 50000-60000, etc.).	1000-9999
Input Path	Directory on FTP host from which to retrieve files. An optional search pattern may be included with the directory and will be evaluated as either a DOS pattern (e.g: a* or a???), or as a Java Regular Expression (e.g. ^a.* or ^a...) depending on the setting of the 'Wildcard pattern' field.	/qa/edaldap/In
Scan Subdirectories	If set look through all sub directories for matching entries.	false Pick one
Destination	Directory on the FTP server into which output files are stored. If missing the home directory of the user used to authenticate to the FTP server is used. Specific file name is optional. Use * in file name to be replaced by timestamp, # by sequential counter. If it is unclear whether path names a directory or a filename, iSM will assume the path names a file.	/qa/edaldap/Out

Obtaining Configuration Properties Using a Document-Centric Query

Using iWay Functional Language (iFL), the internal server provides functions that can address the parsed form of a payload. These functions require a document to be present and in process. This differs from other iFL functions such as _sreg() or _property(), which can access information without a document being present or in process. For XML, the standard Xpath language is provided, while for JSON, JsonPath and JsonPointer are offered.

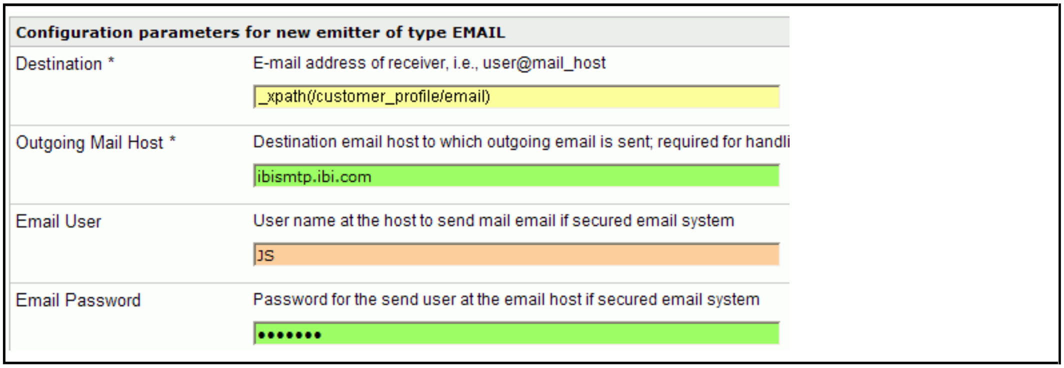
This section describes the use of XPath to route the XML document. However for JSON documents, the _jsonpath() or _jsonptr() functions can be used. For more information on these functions, see the *iWay Functional Language Reference Guide*.

Example: Using XPath to Dynamically Route Output

The <email> node in the following sample code is referenced by iSM to route a reply-to message:

```
<customer_profile>
<name>Joseph Bloggs</name>
<email>Joe_Smith@ibi.com</email>
<vehicles>
   <car>JAGUAR</car>
 </vehicles>
</customer_profile>
```

The XPath notation _xpath(/customer_profile/email) shown in the following image indicates the emitter information derived from the file.



Configuration parameters for new emitter of type EMAIL

Destination *	E-mail address of receiver, i.e., user@mail_host	_xpath(/customer_profile/email)
Outgoing Mail Host *	Destination email host to which outgoing email is sent; required for handli	ibismtp.ibi.com
Email User	User name at the host to send mail email if secured email system	JS
Email Password	Password for the send user at the email host if secured email system	•••••••

Obtaining Configuration Properties Using Special Registers

Special registers are named tokens that contain information available to services. You can use special registers in:

- Incoming documents
- Parameters
- Configuration values

Special registers are accessed by their token name, _sreg(*sregname*).

Some special registers are completed automatically by iWay Service Manager (iSM) during operation. For example, the correlation ID of a message from a queue (for example, JMS, MQ, or SONIC) can be obtained by _sreg(correlid). An emitter with the property correlid= that is used to set the correlation ID might be set to _sreg(msgid) so that the reply is correlated with the incoming message.

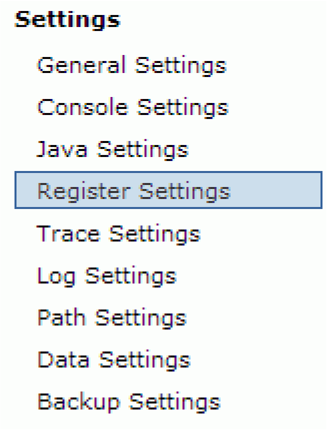
You can define other special registers to the iSM configuration by using the defined special register facility. You can also use a defined special register elsewhere in the configuration and as a value substitution in a document. This feature enables you to define a value, such as a queue manager name, in one place and reference it in another. A system with a dozen MQSeries listeners would then need to change the queue manager name only in one place.

In addition, you can use a special register in the LDAP function as a value for the filter, enabling a search on a complex name without having to spell it out in several places.

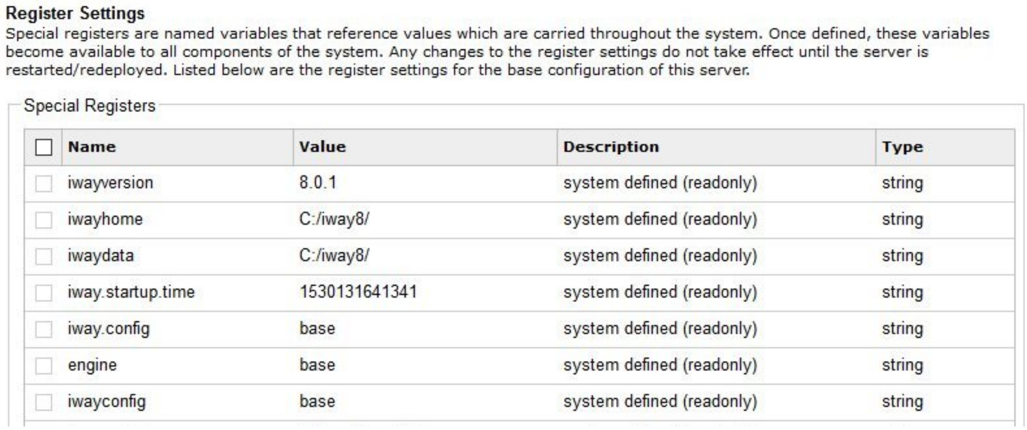
Procedure: How to Display System-Wide Special Registers

To display system-wide special register names and values:

1. In the top pane, click *Server*.
2. From the Settings list in the left pane, click *Register Settings*.



The Register Settings pane opens.



Register Settings

Special registers are named variables that reference values which are carried throughout the system. Once defined, these variables become available to all components of the system. Any changes to the register settings do not take effect until the server is restarted/redeployed. Listed below are the register settings for the base configuration of this server.

Special Registers

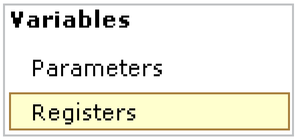
□	Name	Value	Description	Type
□	iwayversion	8.0.1	system defined (readonly)	string
□	iwayhome	C:/iway8/	system defined (readonly)	string
□	iwaydata	C:/iway8/	system defined (readonly)	string
□	iway.startup.time	1530131641341	system defined (readonly)	string
□	iway.config	base	system defined (readonly)	string
□	engine	base	system defined (readonly)	string
□	iwayconfig	base	system defined (readonly)	string

The name, value, and data type for the special registers in use are displayed.

Procedure: How to Display Special Registers (Register Sets) for a Channel

To display special registers (register sets) for a channel:

1. In the top pane, click *Registry*.
2. From the Variables list in the left pane, click *Registers*.

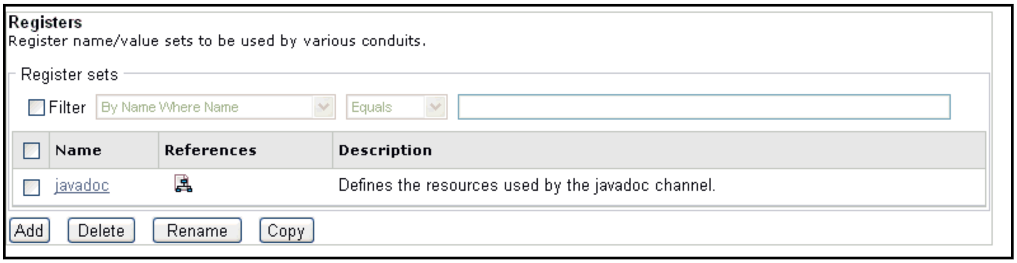


The Registers pane opens.

The special register sets used by various conduits are displayed in this pane. In this example, the special register set for a channel called javadoc is shown.

You can use the Registers pane to define groups of registers that can be assigned to channels and process flows (in iIT Designer). These registers may be used to configure components of a channel and process flow, or used by these components at run time in the case when components refer to registers.

3. In the Registers pane, click *javadoc*.



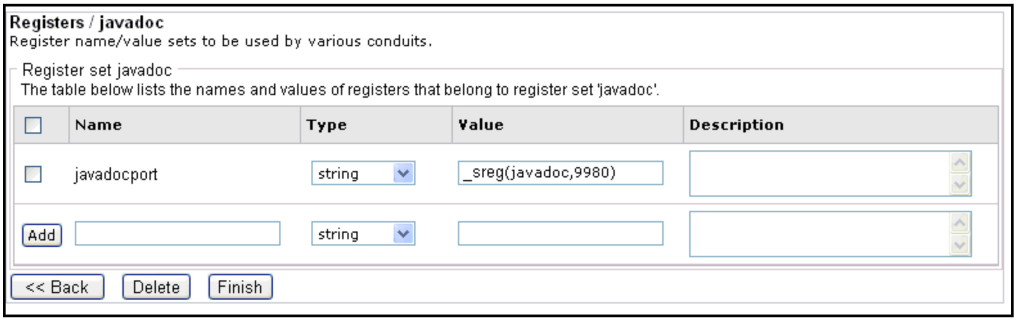
Registers
Register name/value sets to be used by various conduits.

Register sets

□ Filter | By Name Where Name | Equals

□	Name	References	Description
□	javadoc		Defines the resources used by the javadoc channel.

The Register set pane for the javadoc channel opens.



Registers / javadoc
Register name/value sets to be used by various conduits.

Register set javadoc
The table below lists the names and values of registers that belong to register set 'javadoc'.

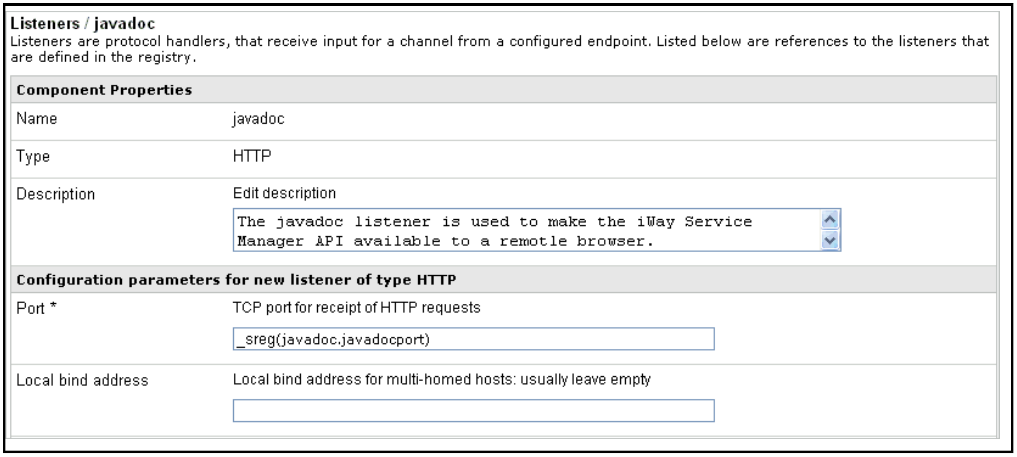
□	Name	Type	Value	Description
□	javadocport	string	_sreg(javadoc,9980)	
Add		string		

This table lists the names, types, values, and descriptions of registers that belong to a register set. For example, javadocport is a register that has been defined for the javadoc register set.

To see where the javadocport register is actually being used, you can look at the inlet of the javadoc channel, which is an HTTP listener.

To use the register set for a channel, the SREG definition must fully qualify the register set, for example, _sreg(javadoc.javadocport). Notice the period that separates the register set from the actual register name defined within the register set.

The following image shows the Listeners pane for the javadoc listener.



Listeners / javadoc

Listeners are protocol handlers, that receive input for a channel from a configured endpoint. Listed below are references to the listeners that are defined in the registry.

Component Properties	
Name	javadoc
Type	HTTP
Description	Edit description The javadoc listener is used to make the iWay Service Manager API available to a remotle browser.
Configuration parameters for new listener of type HTTP	
Port *	TCP port for receipt of HTTP requests _sreg(javadoc.javadocport)
Local bind address	Local bind address for multi-homed hosts: usually leave empty

Notice the Port field, which contains the following value:

```
_sreg(javadoc.javadocport)
```

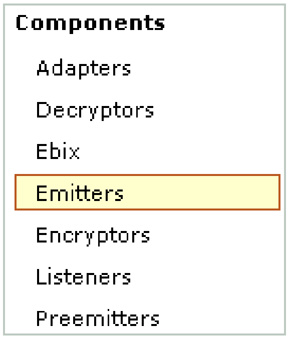
Procedure: How to Use a Special Register

This procedure uses a special register to configure an emitter. For example, to send reply mail to the sender, you can configure an email emitter as follows:

```
Destination=_sreg(from)
```

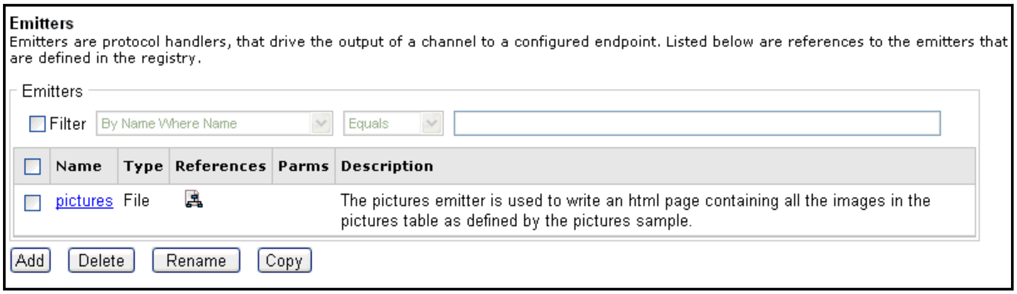
To use a special register:

1. In the top pane, click *Registry*.
2. From the Components list in the left pane, click *Emitters*.



The Emitters pane opens.

3. Click *Add*.



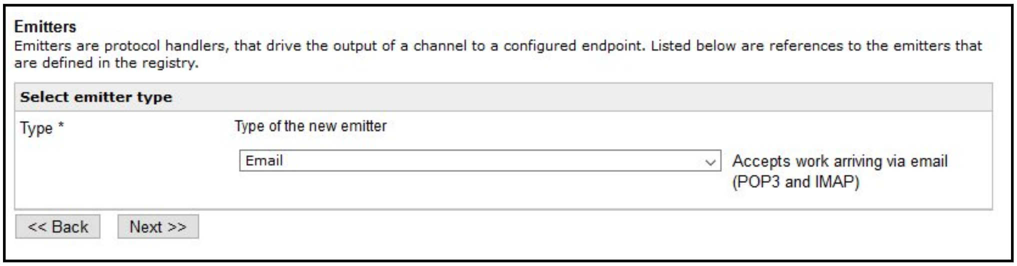
Emitters

Emitters are protocol handlers, that drive the output of a channel to a configured endpoint. Listed below are references to the emitters that are defined in the registry.

□	Name	Type	References	Parms	Description
□	pictures	File			The pictures emitter is used to write an html page containing all the images in the pictures table as defined by the pictures sample.

The Select emitter type pane opens.

4. From the Type drop-down list, select *EMAIL* and click *Next*.



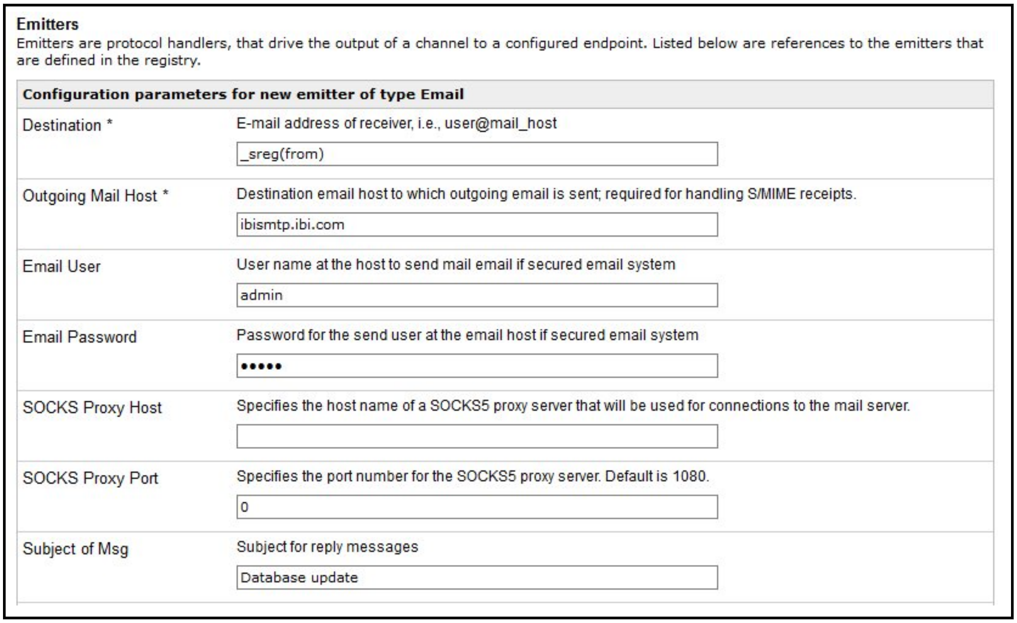
Emitters

Emitters are protocol handlers, that drive the output of a channel to a configured endpoint. Listed below are references to the emitters that are defined in the registry.

Select emitter type

Type *	Type of the new emitter Email	Accepts work arriving via email (POP3 and IMAP)

Configuration parameters for the email emitter are displayed.



Emitters

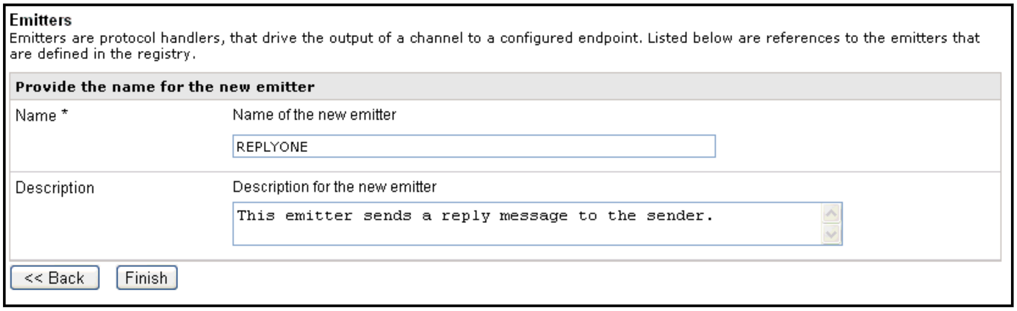
Emitters are protocol handlers, that drive the output of a channel to a configured endpoint. Listed below are references to the emitters that are defined in the registry.

Configuration parameters for new emitter of type Email

Destination *	E-mail address of receiver, i.e., user@mail_host _sreg(from)
Outgoing Mail Host *	Destination email host to which outgoing email is sent; required for handling S/MIME receipts. ibismtp.ibi.com
Email User	User name at the host to send mail email if secured email system admin
Email Password	Password for the send user at the email host if secured email system •••••
SOCKS Proxy Host	Specifies the host name of a SOCKS5 proxy server that will be used for connections to the mail server.
SOCKS Proxy Port	Specifies the port number for the SOCKS5 proxy server. Default is 1080. 0
Subject of Msg	Subject for reply messages Database update

5. In the Destination field, type *_sreg(from)*.
6. In the Outgoing Mail Host field, specify an email host.
7. Enter appropriate values for the remaining parameters that are applicable to your environment, for example, an email user name and password, and click *Next*.

   The following pane, which contains a Name and Description field for the emitter is displayed.



Emitters

Emitters are protocol handlers, that drive the output of a channel to a configured endpoint. Listed below are references to the emitters that are defined in the registry.

Provide the name for the new emitter

Name * — Name of the new emitter: REPLYONE

Description — Description for the new emitter: This emitter sends a reply message to the sender.

8. Type a name and description for the emitter, and click *Finish*.

   Once executed, this emitter will send an email message to the recipient using the address defined in the special register _sreg(from).

Enriching a Document With the Content of a Special Register

You can enrich a document by referring to a special register that supplies content for the document. For example, you can update the following document with the name of the sender of the email message.

```
<document>
 <sender></sender>
</document>
```

To place the name of the sender of the email in the document code:

```
<document>
<sender>_sreg(from)</sender>
</document>
```

After execution, the document is enriched as follows:

```
<document>
<sender>John_Smith@iWaysoftware.com</sender>
</document>
```

Processing a document using this method requires that the EvalWalk service be configured and chained. For more information, see the *iWay Service Manager Component and Functional Language Reference Guide*.

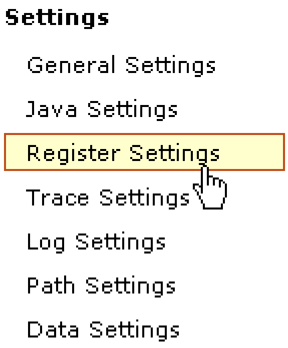
Using Registers, Register Sets, and Parameters

This section explains the differences between registers and register sets in iWay Service Manager and describes how to bind register sets to a channel. In addition, information on how to define parameters is provided.

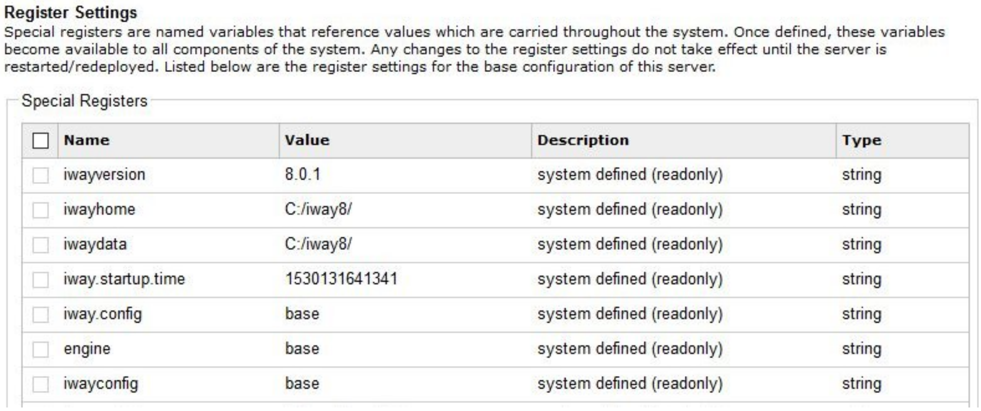
Registers

Registers are global parameter values that are available to any process running within an iWay Service Manager instance. These values are set in the runtime environment and can be used by any iWay Registry component, for example, listener, process flow, adapter, agent, and so on.

The name and value of a register can be configured in the Register Settings pane of the iWay Service Manager Administration Console. To access the Register Settings pane, click *Register Settings* from the Settings list in the left pane of the Server page, as shown in the following image.



The Register Settings pane opens.



Register Settings

Special registers are named variables that reference values which are carried throughout the system. Once defined, these variables become available to all components of the system. Any changes to the register settings do not take effect until the server is restarted/redeployed. Listed below are the register settings for the base configuration of this server.

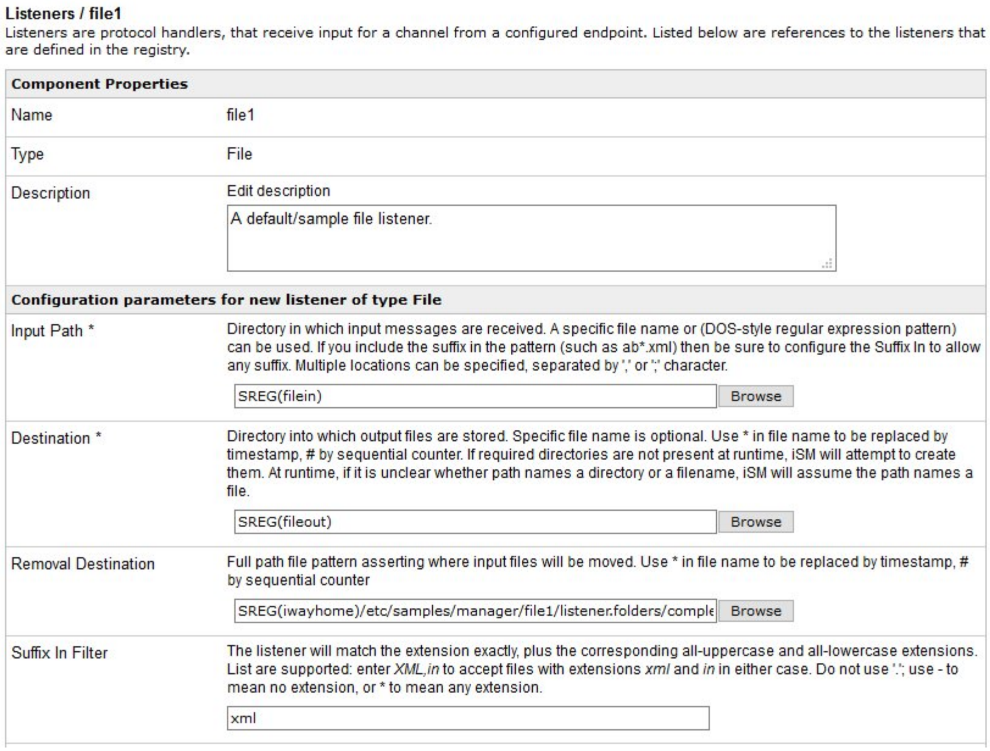
Special Registers

☐	Name	Value	Description	Type
☐	iwayversion	8.0.1	system defined (readonly)	string
☐	iwayhome	C:/iway8/	system defined (readonly)	string
☐	iwaydata	C:/iway8/	system defined (readonly)	string
☐	iway.startup.time	1530131641341	system defined (readonly)	string
☐	iway.config	base	system defined (readonly)	string
☐	engine	base	system defined (readonly)	string
☐	iwayconfig	base	system defined (readonly)	string

The name, value, description, and data type for the available registers are displayed.

Notice the Special Register (SREG) in the list called *iwayconfig*, which is created by default for each iWay Service Manager configuration that is created. For the master configuration called *base*, the _sreg(iwayconfig) would have a value equal to *base*.

The following image shows the configuration parameters for a listener called *file1*. The Input Path field contains an SREG called *filein* to represent the input path and the Destination field contains an SREG called *fileout* to represent the output path.



Listeners / file1

Listeners are protocol handlers, that receive input for a channel from a configured endpoint. Listed below are references to the listeners that are defined in the registry.

Component Properties	
Name	file1
Type	File
Description	Edit description A default/sample file listener.
Configuration parameters for new listener of type File	
Input Path *	Directory in which input messages are received. A specific file name or (DOS-style regular expression pattern) can be used. If you include the suffix in the pattern (such as ab*.xml) then be sure to configure the Suffix In to allow any suffix. Multiple locations can be specified, separated by ',' or ';' character. SREG(filein) Browse
Destination *	Directory into which output files are stored. Specific file name is optional. Use * in file name to be replaced by timestamp, # by sequential counter. If required directories are not present at runtime, iSM will attempt to create them. At runtime, if it is unclear whether path names a directory or a filename, iSM will assume the path names a file. SREG(fileout) Browse
Removal Destination	Full path file pattern asserting where input files will be moved. Use * in file name to be replaced by timestamp, # by sequential counter SREG(iwayhome)/etc/samples/manager/file1/listener.folders/comple Browse
Suffix In Filter	The listener will match the extension exactly, plus the corresponding all-uppercase and all-lowercase extensions. List are supported: enter *XML,in* to accept files with extensions *xml* and *in* in either case. Do not use '.'; use - to mean no extension, or * to mean any extension. xml

_sreg(filein) and _sreg(fileout) provide good examples of how registers can be used in iWay Service Manager once they are configured in the Register Settings pane.

Notice that the Removal Destination field contains the following directory path suffix:

```
_sreg(iwayhome)/etc/samples/manager/file1/listener.folders/complete
```

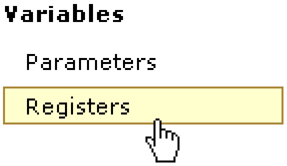
You can also use multiple SREGs to create a path, for example, *_sreg(homedir)* or *_sreg(userdir)*.

Note: After a new Special Register is created, it is immediately available for use by all iWay Registry components, for example, listener, process flow, adapter, agent, and so on. However, you must perform an iWay Service Manager console restart (warm restart) after editing the value of a Special Register if you want the new value to be used by any associated iWay Registry components.

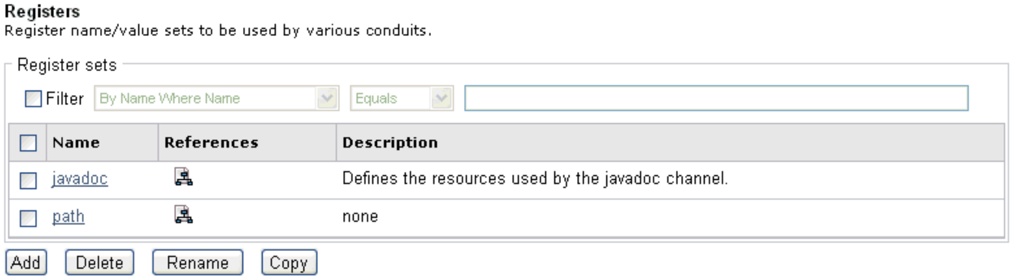
Register Sets

Register sets are very similar to registers, but they are not global by default. A register set contains a pool of one or more Special Registers (SREGs).

Register sets can be configured in the Registers pane of the iWay Service Manager Administration Console. To access the Registers pane, click *Registers* from the Variables list in the left pane of the Registry page, as shown in the following image.



The Registers pane opens.



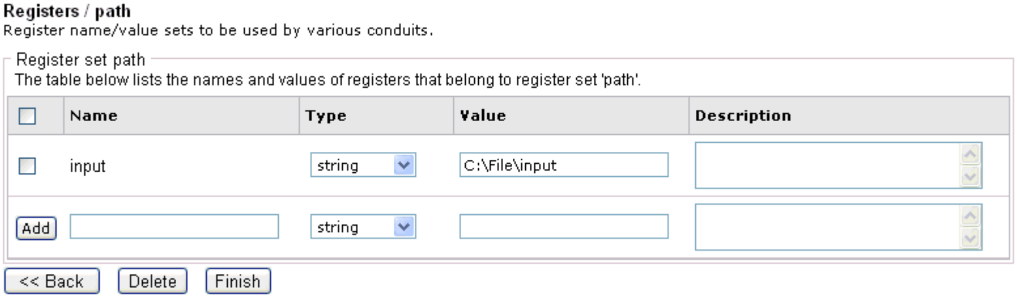
Registers

Register name/value sets to be used by various conduits.

□	Name	References	Description
□	javadoc		Defines the resources used by the javadoc channel.
□	path		none

The name, reference, and description for the available register sets are displayed. A register set called *javadoc* is created by default. For demonstration purposes, a new register set called *path* has been created. You can click the name of a register set to open a configuration pane for the register set.

For example, the following image shows the Registers / path configuration pane. The name of the register set is *path* and the name of the register that belongs to this register set is *input*. Using this register set configuration pane, you can add multiple registers to a register set as required.



Registers / path
Register name/value sets to be used by various conduits.

Register set path
The table below lists the names and values of registers that belong to register set 'path'.

□	Name	Type	Value	Description
□	input	string	C:\File\input	
Add		string		

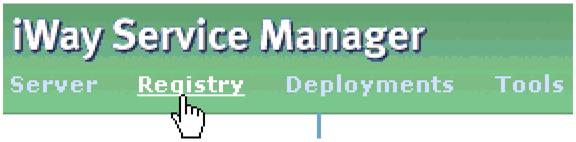
<< Back   Delete   Finish

It is important to understand that register sets are used local to a channel and are defined within the component, which is different from an SREG. This is how you must use the register set _sreg(path.input). Also note that in order to use a register set within a channel you must bind the register set to that channel.

Procedure: How to Bind a Register Set to a Channel

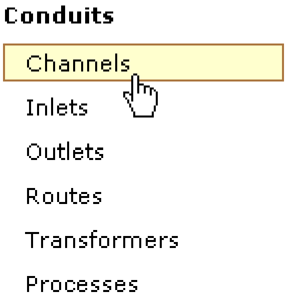
To bind a register set to a channel:

1. Click *Registry* in the menu bar, which is located in the top pane.

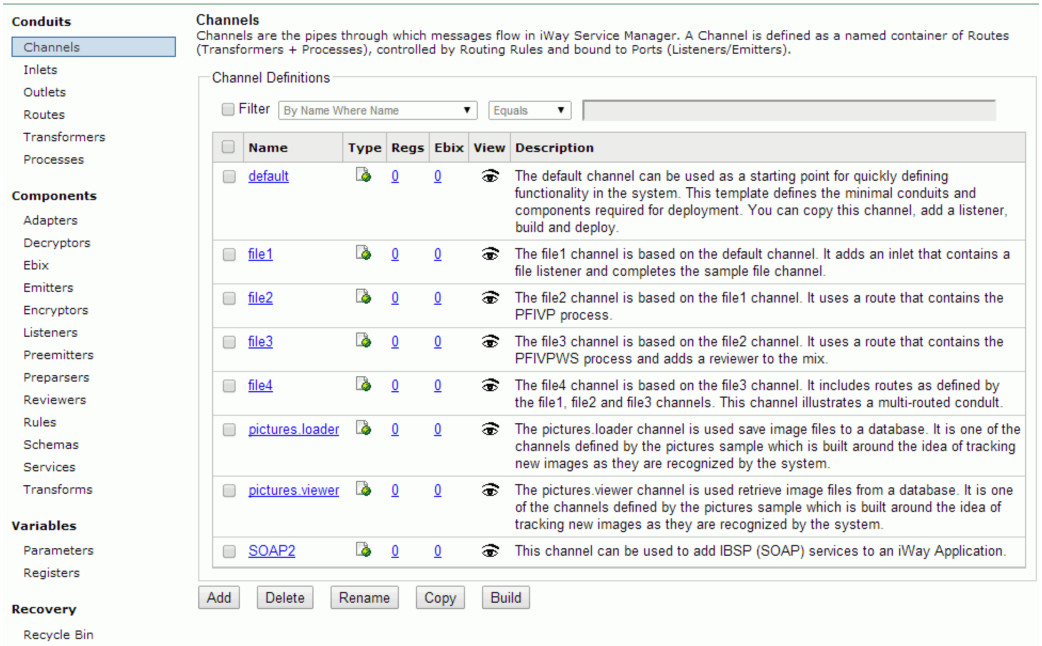


The Registry - Repository pane opens, showing links to various types of conduits and components you can configure.

2. Click *Channels*.



The Channels pane opens.



Channels

Channels are the pipes through which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

Channel Definitions

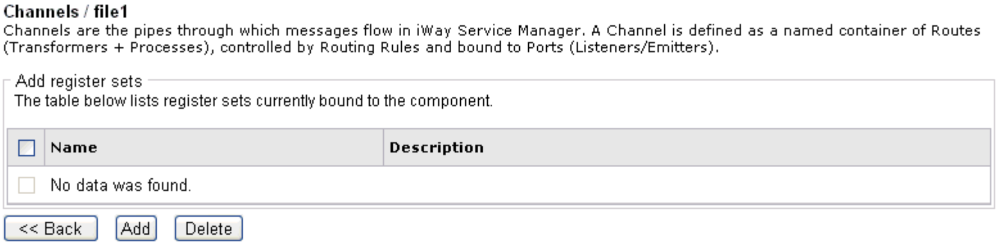
□ Filter By Name Where Name ▼ Equals ▼

□	Name	Type	Regs	Ebix	View	Description
□	default		0	0		The default channel can be used as a starting point for quickly defining functionality in the system. This template defines the minimal conduits and components required for deployment. You can copy this channel, add a listener, build and deploy.
□	file1		0	0		The file1 channel is based on the default channel. It adds an inlet that contains a file listener and completes the sample file channel.
□	file2		0	0		The file2 channel is based on the file1 channel. It uses a route that contains the PFIVP process.
□	file3		0	0		The file3 channel is based on the file2 channel. It uses a route that contains the PFIVPWS process and adds a reviewer to the mix.
□	file4		0	0		The file4 channel is based on the file3 channel. It includes routes as defined by the file1, file2 and file3 channels. This channel illustrates a multi-routed conduit.
□	pictures.loader		0	0		The pictures.loader channel is used save image files to a database. It is one of the channels defined by the pictures sample which is built around the idea of tracking new images as they are recognized by the system.
□	pictures.viewer		0	0		The pictures.viewer channel is used retrieve image files from a database. It is one of the channels defined by the pictures sample which is built around the idea of tracking new images as they are recognized by the system.
□	SOAP2		0	0		This channel can be used to add IBSP (SOAP) services to an iWay Application.

The Regs column (highlighted) shows the number of register sets that are currently bound to each available channel.

3. In the Regs column, click the number for a channel, for example, *file1*.

The Add register sets pane opens for the file1 channel, as shown in the following image.



Channels / file1

Channels are the pipes through which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

Add register sets

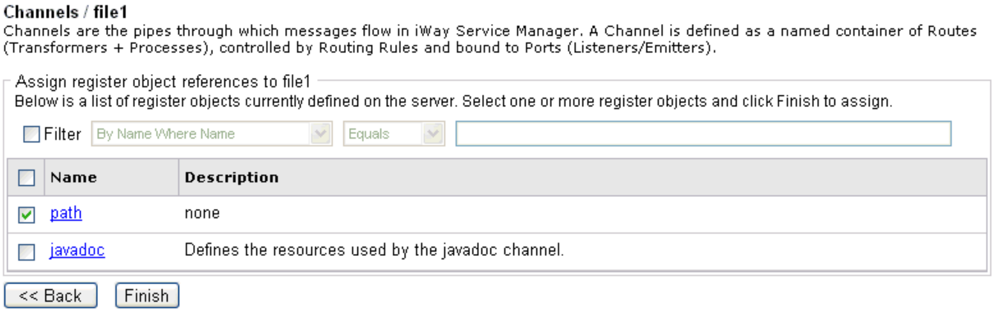
The table below lists register sets currently bound to the component.

□	Name	Description
□	No data was found.	

Notice that there are currently no register sets bound to the file1 channel.

4. Click *Add*.

   The Assign register object references to file1 pane opens and provides a list of available register sets.



Channels / file1

Channels are the pipes through which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

Assign register object references to file1

Below is a list of register objects currently defined on the server. Select one or more register objects and click Finish to assign.

□ Filter By Name Where Name Equals

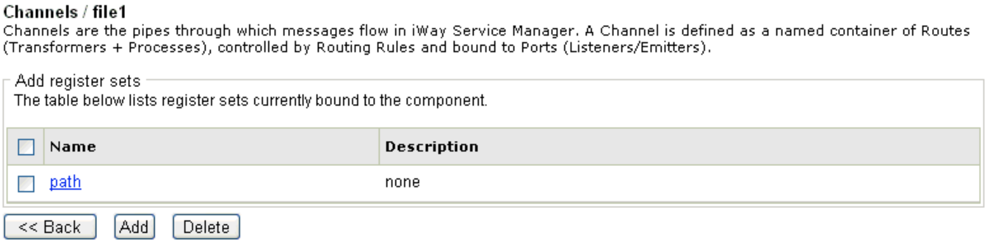
□	Name	Description
☑	path	none
□	javadoc	Defines the resources used by the javadoc channel.

Note: Clicking on the name of a register set opens the configuration pane that allows you to modify, add, or remove registers for that register set.

5. Select the check box next to the register set you want to add, for example, *path*.

6. Click *Finish*.

   You are returned to the Add register sets pane for the file1 channel.



Channels / file1

Channels are the pipes through which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

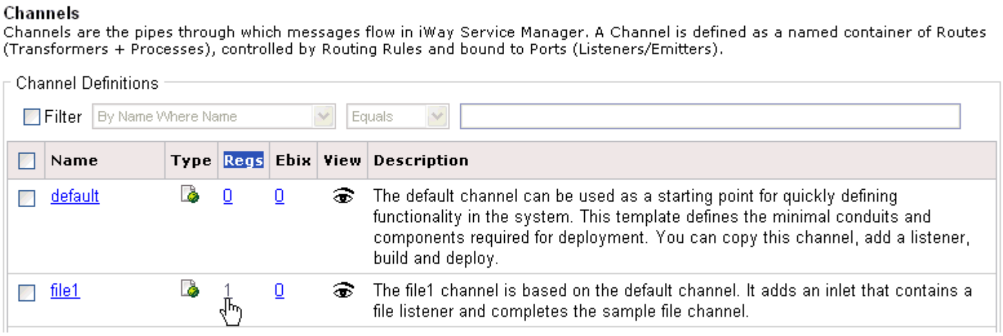
Add register sets

The table below lists register sets currently bound to the component.

□	Name	Description
□	path	none

Notice that the register set called *path* is now listed for the file1 channel.

When you return to the Channels pane, notice the number 1 that appears in the Regs column for the file1 channel, as shown in the following image.



Channels

Channels are the pipes through which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

Channel Definitions

□ Filter By Name Where Name | Equals

□	Name	Type	Regs	Ebix	View	Description
□	default		0	0		The default channel can be used as a starting point for quickly defining functionality in the system. This template defines the minimal conduits and components required for deployment. You can copy this channel, add a listener, build and deploy.
□	file1		1	0		The file1 channel is based on the default channel. It adds an inlet that contains a file listener and completes the sample file channel.

Note: You must redeploy the channel if changes were made to a register set you are using.

You have successfully bound a register set to a channel in iWay Service Manager.

Defining Parameters

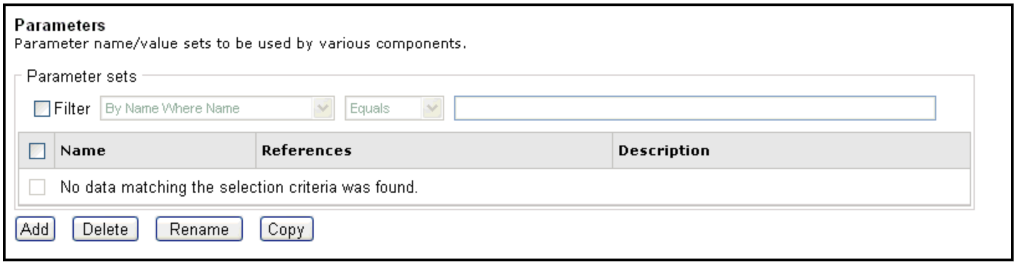
The iWay Service Manager Administration Console allows you to define sets of parameters that can be assigned to certain emitters and/or exits. These are dynamic, user-defined parameters that are not described by the metadata of an object. For example, they can be used to provide additional protocol headers.

Procedure: How to Define Parameters

To define parameters:

1. In the left console pane of the Registry menu, select *Parameters*.

The Parameters pane opens.



Parameters
Parameter name/value sets to be used by various components.

Parameter sets

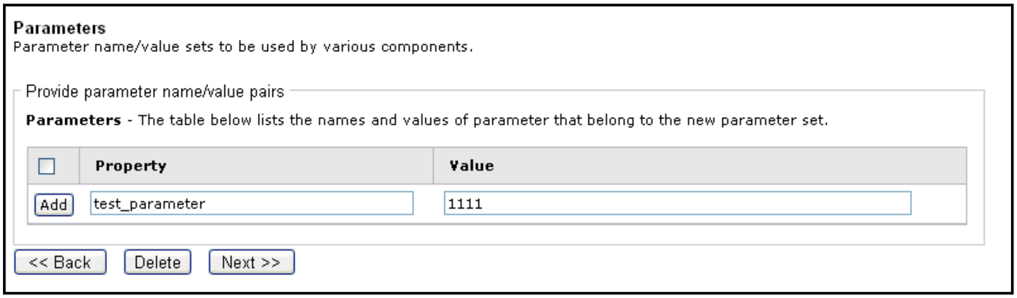
☐ Filter By Name Where Name | Equals |

☐	Name	References	Description
☐	No data matching the selection criteria was found.		

The parameter sets used by various conduits are displayed in this pane. Currently, there are no parameter sets defined.

2. Click *Add*.

   The Provide parameter name/value pairs pane opens.



Parameters
Parameter name/value sets to be used by various components.

Provide parameter name/value pairs

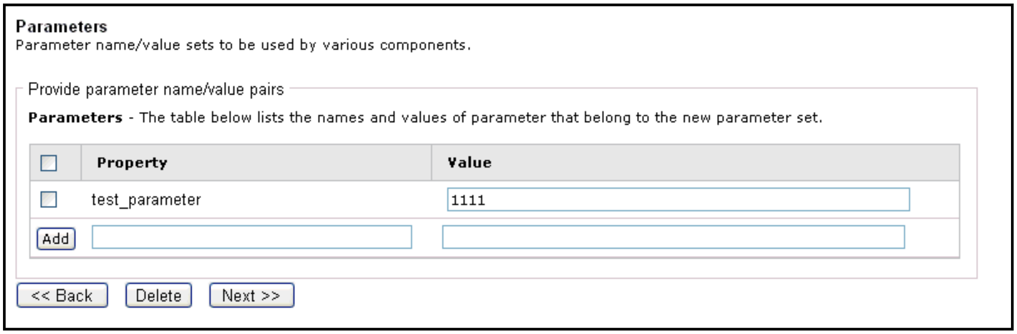
Parameters - The table below lists the names and values of parameter that belong to the new parameter set.

☐	Property	Value
Add	test_parameter	1111

The table that is provided lists the names and values of parameters that belong to the new parameter set.

3. Specify a parameter and a corresponding value.
4. Click *Add*.

   The parameter is added, as shown in the following image.



Parameters
Parameter name/value sets to be used by various components.

Provide parameter name/value pairs

Parameters - The table below lists the names and values of parameter that belong to the new parameter set.

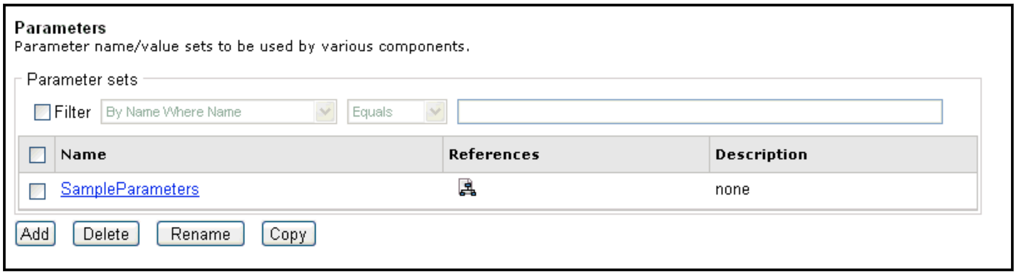
☐	Property	Value
☐	test_parameter	1111
Add		

You can add as many parameters as required.

5. Click *Next*.

6. Provide a name and, optionally, a description, for the parameter set, and click *Finish*.

   You are returned to the main Parameters pane, which now includes the new parameter set that was defined.



Parameters
Parameter name/value sets to be used by various components.

Parameter sets

□ Filter By Name Where Name Equals

□	Name	References	Description
□	SampleParameters		none

Chapter 5

Managing Configurations

This section describes how to add multiple configurations or users to iWay Service Manager, and manage configurations in batch.

In this chapter:

Introducing iWay Service Manager Configurations

Each instance of iWay Service Manager (iSM) is called a managed server. You can create multiple managed servers. Each configuration has a name, which you can access through the iWay Service Manager Administration Console.

Each configuration is defined, started, and stopped independently. However, the iWay Administration Services Console is used to control them all. You can have one or more configurations operating simultaneously within iSM. Each operates as its own process and does not depend upon or affect any other configuration. You must avoid conflicts that may arise by incorrectly defining configurations, for example, using the same file directories or TCP ports.

You must configure each instance for the task it is to accomplish. For example, you might have a production instance and a test instance. You can develop elements used in iSM to process messages in the iWay design-time tools, and then deploy them to the appropriate instance for execution. You can test the message in the test instance and then deploy the software and configuration to the production instance.

Management Option

The Server Management pane enables you to add configurations or users to iSM. Through this pane you can add users (administrators and non-administrators) with read/write or monitor access to specific configurations.

Configuration management enables you to:

- Begin from a single configuration that is easy to understand.
- Manage this configuration and run it as a single tailored configuration.
- At the push of a button, generate an additional configuration based on an existing configuration.
- Use drop-down lists of all available configurations, by logical name, to implement as templates.
- Delete a configuration, which automatically removes everything associated with creating the configuration.
- Add components to a newly created configuration, through the use of named packages.
- Give privileges to or delete existing users or create new users using a global interface.

Creating a Configuration

iWay Service Manager provides model configurations that you can use as is, or edit to create a custom configuration. When you install iSM, it includes the following configurations:

- **raw.** This is an empty configuration that has no defined message processing capabilities. It is provided as a template for creating a configuration from scratch.

  You must never modify the raw configuration. It is reserved as the bootstrap configuration for creating other configurations.

- **base.** This configuration has limited capabilities to handle some messages for SOAP activity. It hosts an information window about iSM. This configuration is required to complete the registration process.

You can base your new configuration on one of these two model configurations or on another existing configuration. To avoid conflicts with other configurations, the system automatically assigns the next available port as the console port for the new configuration. This port is used for internal communication and cannot be accessed directly. You can override this value. If you override the port with a value already assigned to another configuration, then you are prompted to make a different selection.

Based on the template you use, the configuration software generates the new configuration and the required folders and files. You can also add other components to your configuration. Before you can use a configuration effectively, you must define its channels and business logic.

To create a configuration:

1. Add the configuration to the console.
2. Create a Windows service for the new configuration.
3. Start the Windows service for the new configuration.

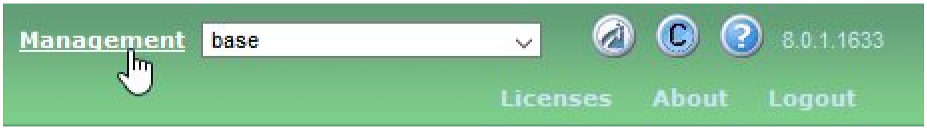
In order to take advantage of performance enhancements, Service Manager configurations can be installed to run as services in separate Java processes. In order to do this:

1. Uninstall the configuration that you wish to modify.
2. Create a service for the new configuration with Java running in a separate process. This is explained in *How to Create a Windows Service for a New Configuration* on page 95.
3. Start the service for the new configuration.

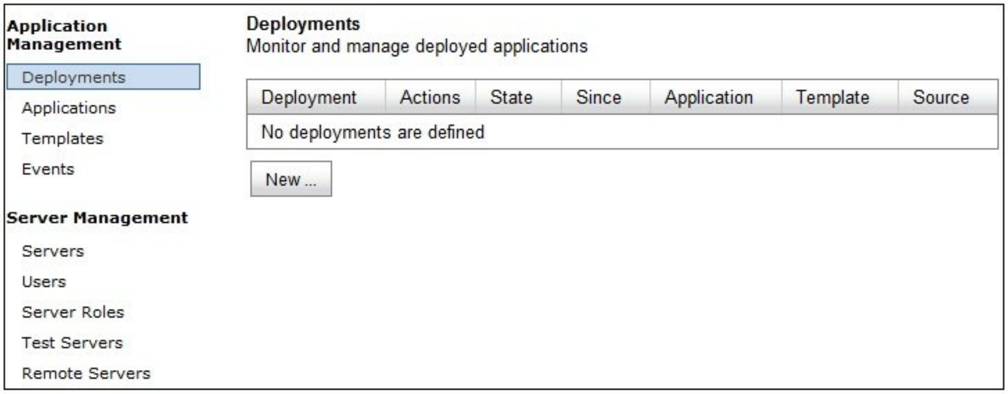
Procedure: How to Add a Configuration to iWay Service Manager

To add a configuration to iWay Service Manager:

1. In the top pane, click *Management*.



The Deployments pane opens by default, as shown in the following image.

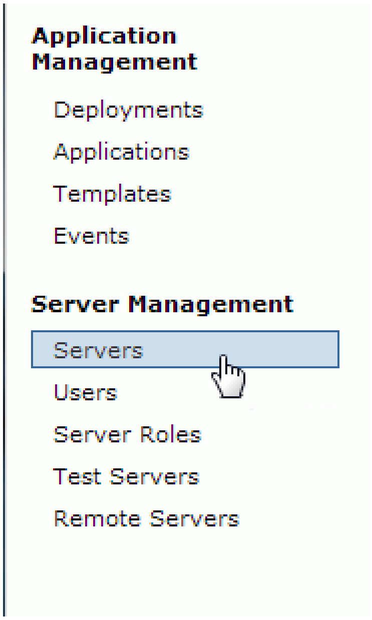


Deployments

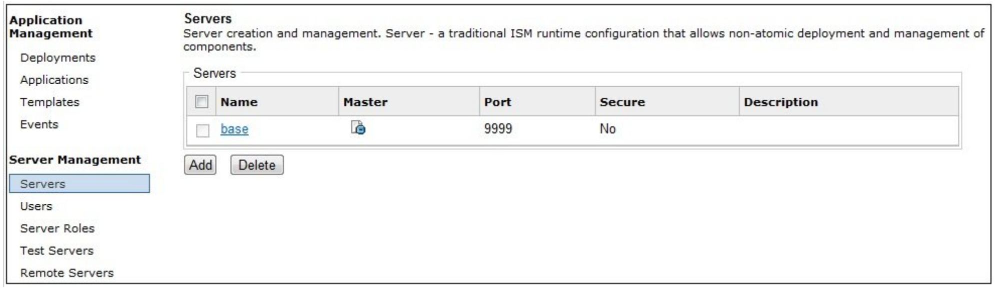
Monitor and manage deployed applications

Deployment	Actions	State	Since	Application	Template	Source
No deployments are defined						

2. In the Server Management section, click *Servers*, as shown in the following image.



The Servers pane opens, as shown in the following image.



Servers

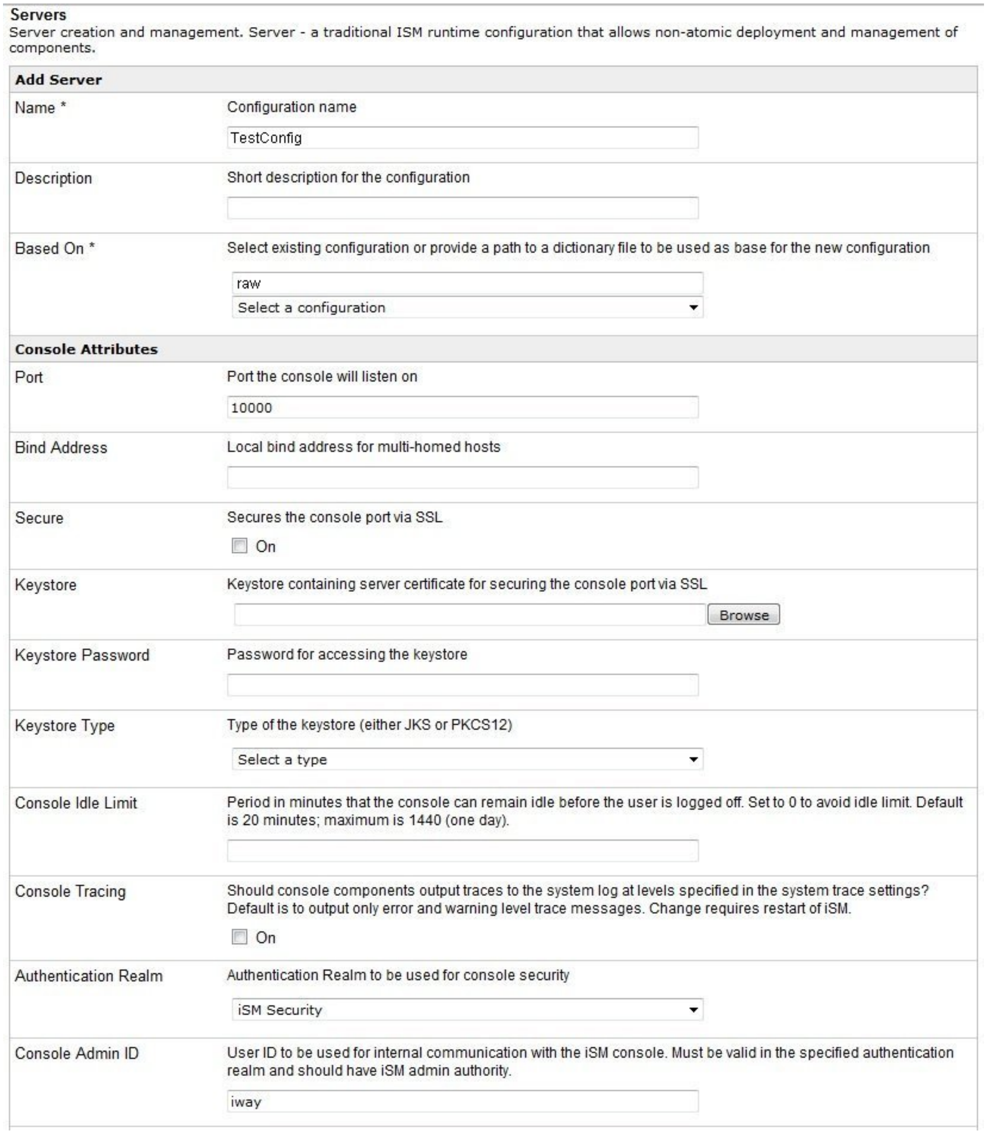
Server creation and management. Server - a traditional ISM runtime configuration that allows non-atomic deployment and management of components.

Servers

□	Name	Master	Port	Secure	Description
□	base		9999	No	

3. Click *Add*.

The Servers configuration pane opens, as shown in the following image.



Servers

Server creation and management. Server - a traditional ISM runtime configuration that allows non-atomic deployment and management of components.

Add Server		
Name *	Configuration name	TestConfig
Description	Short description for the configuration	
Based On *	Select existing configuration or provide a path to a dictionary file to be used as base for the new configuration	raw Select a configuration ▾
Console Attributes		
Port	Port the console will listen on	10000
Bind Address	Local bind address for multi-homed hosts	
Secure	Secures the console port via SSL	□ On
Keystore	Keystore containing server certificate for securing the console port via SSL	Browse
Keystore Password	Password for accessing the keystore	
Keystore Type	Type of the keystore (either JKS or PKCS12)	Select a type ▾
Console Idle Limit	Period in minutes that the console can remain idle before the user is logged off. Set to 0 to avoid idle limit. Default is 20 minutes; maximum is 1440 (one day).	
Console Tracing	Should console components output traces to the system log at levels specified in the system trace settings? Default is to output only error and warning level trace messages. Change requires restart of iSM.	□ On
Authentication Realm	Authentication Realm to be used for console security	iSM Security ▾
Console Admin ID	User ID to be used for internal communication with the iSM console. Must be valid in the specified authentication realm and should have iSM admin authority.	iway

a. In the *Name* field (required), type a name for the configuration, for example, TestConfig.

b. In the *Description* field (optional), type a description for the configuration, for example, Configuration for testing.

c. In the *Based On* field (required), select an existing configuration from the drop-down list, or provide and absolute path to a dictionary file.

d. In the *Port* field under Console Attributes, type the port number on which the console is listening.

e. In the *Bind Address* field (optional), provide a local bind address for multi-homed hosts.

f. If you want to secure the console port via SSL, select the *On* check box (optional) in the Secure section.

g. In the *Keystore* field, type the keystore pathway containing the server certificate for securing the console port using SSL.

h. In the *Keystore Password* field, type the password for accessing the keystore.

i. In the *Keystore Type* drop-down list, select the keystore type (either JKS or PKCS12).

j. In the *Console Idle Limit* field, type the period in minutes that the console can remain idle before the user is logged off. Default is 20 minutes.

k. In the *Console Tracing* field, check *On* to enable component output traces.

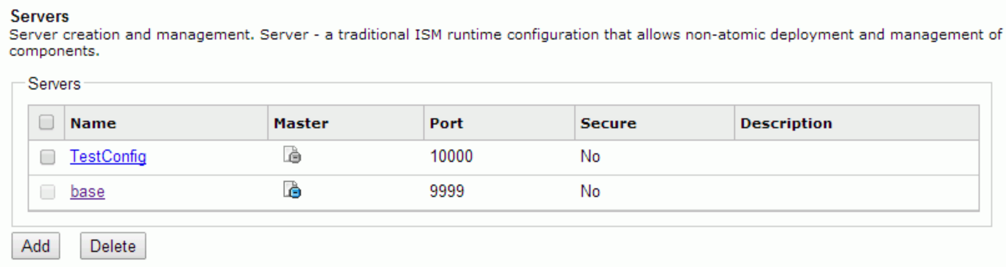
l. In the *Authentication Realm* drop-down list, select the Authentication Realm to be used for console security.

m. In the *Console Admin ID* field, type the User ID to be used for internal communication with the iSM console. The ID must be valid in the specified authentication realm and should have iSM admin authority.

n. In the *Console Admin Password* field, type the password for the console admin account.

4. Click *Finish*.

The new configuration, TestConfig, appears in the Configurations list, as shown in the following image.



Servers

Server creation and management. Server - a traditional ISM runtime configuration that allows non-atomic deployment and management of components.

Servers

□	Name	Master	Port	Secure	Description
□	TestConfig		10000	No	
□	base		9999	No	

After your configuration is created, you can create a Windows service to start the configuration. This is optional.

Procedure: How to Create a Windows Service for a New Configuration

To create a Windows service that can be used to start a new configuration:

1. Open a Command Prompt window and navigate to the iWay home bin directory. For example, on Windows, if iWay is installed in C:\Program Files\iWay8, go to

   ```
   C:\Program Files\iWay8\bin
   ```

   The *iwsrv* command starts iSM in a command window and allows you to create a Windows service.

2. At the command prompt, type

   ```
   iwsrv config_name -s install
   ```

   To create a service with Java running in a separate process, type

   ```
   iwsrv config_name -s install -l java
   ```

   where:

   config_name

   Is the name of the configuration for which you are creating a service.

   A message appears, indicating that the service was installed successfully.

   Your next task is to start the configuration as a service. For more information, see *How to Start a Configuration as a Service on Windows* on page 97.

Procedure: How to Manage a Configuration With IWSRV

The full syntax for the *iwsrv* command is:

```
iwsrv [configuration] [-s service] [-l launch] [options]
```

where:

configuration

Is the name of the server configuration that is loaded for this instance. The default value is base.

service

Is the name of the service that is executed. Valid values are:

start: Starts the server configuration (default).

stop: Stops the server configuration.

install: Installs the server configuration.

remove: Removes the server configuration.

query: Queries the server configuration.

launch

Specifies the launch method. Valid methods are:

java: Loads Java in a separate process and uses the JVM options, NT dependencies, and other preferences found within the iSM configuration that are configured through the console. For example: `iwsrv.exe base -s start -l java`

script file: Specifies a script file that defines the run-time preferences. This script file must be located in the iWay Service Manager installation directory. For example: `iwsrv.exe base -s start -l iWay8.cmd`

Both of the above uses of -l will force the service to load Java in a separate process. When the service is stopped, both iwsrv.exe and java.exe are terminated.

options

Specifies tracing or server back-up information. Valid values include:

-b: Indicates that Service Manager is a back-up server, for example:

```
iwsrv.exe base -s start -b
```

-c: Turns tracing on. In this mode, you can display useful error messages on the console. For example, you can display a message that says the Java Runtime Environment (JRE) is not properly installed. For example:

```
iwsrv.exe base -s start -c
```

-d: Limits tracing to debug only, for example:

```
iwsrv.exe base -s start -d
```

-f: [PATH] filters the system path when invoking JAVA. [RESTART] suppresses the JVM fault restart capability.

-h: iWay8 home directory.

-t: The amount of time (in seconds) to process service shutdown.

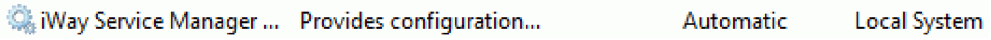
Procedure: How to Start a Configuration as a Service on Windows

To start a new configuration as a service on Windows:

1. From the Start menu, select *Start*, *Settings*, *Control Panel*, *Administrative Tools*, and then *Services*.

   The Services window opens.

2. Scroll down to display the iWay Service Manager services, as shown in the following image.

   iWay Service Manager ... Provides configuration... Automatic Local System

3. Right-click the service you created, for example, iWay Service Manager - TestConfig and then, select *Start* (if it is not already started).

   The service status changes to Started. The default start-up type is Automatic.

Procedure: How to Start a Configuration as a Service on a Non-Windows Platform

To start a new configuration as a service on a platform other than Windows:

1. Locate the default start-up file, which is supplied for the base configuration, in the iWay8/bin directory.

   On UNIX, this file is called:

   ```
   startservice.sh
   ```

   On other non-Windows platforms, the name may vary.

2. Create a copy of the start-up file for your service.

3. Edit the copy of the start-up file using an editor and navigate to the following section in the file to replace *base* with the name of your configuration:

   ```
   #!/bin/sh
   #####################################################################
   #
   # Init
   # Setup global variables and signal handling
   #
   # Edit following lines to point to install dir and change user ID"
   IWAY8=/WorkSource/iWay8/
   IWAYUSER=root
   IWAYCONFIG=baseMOD=""
   ```

4. Navigate to the following section in the file and change the log file name, *serviceOut.txt*, to be unique to this service:

```
cd $IWAY8/config/$IWAYCONFIG
if test 'uname' = 'OS400'; then
     java $REMDBG -cp $CLASSPATH -DIWAY8=$IWAY8
  com.ibi.service.edaqmSilentService -config $IWAYCONFIG >>
  $IWAY8/serviceOut.txt
&elif test 'uname' = 'OS/390'; then
     java $REMDBG -cp $CLASSPATH -DIWAY8=$IWAY8
  com.ibi.service.edaqmSilentService -config $IWAYCONFIG >>
  $IWAY8/serviceOut.txt
&else    su $IWAYUSER -c "java $REMDBG -cp $CLASSPATH -
  Diwaysoftware.af.idocument=com.ibi.edaqm.XDDocument -DIWAY8=$IWAY8
  com.ibi.service.edaqmSilentService -config $IWAYCONFIG >>
  $IWAY8/serviceOut.txt
&"fi
```

5. Save your changes.
6. Execute the start-up file for your service.

   The service is started.

Working With Configurations

You can select and modify any configuration. However, altering the “raw” configuration compromises its usefulness as a template. iWay Software recommends that you reserve the raw configuration as a template for designing other configurations.

To change configurations or users, you must access the iWay Service Manager Administration Console through the master configuration port. This feature prevents simultaneous access to configuration repositories. Otherwise, the master configuration behaves like any other configuration. You can change its properties or assign them to another configuration, at which point you can delete the former master configuration. However, you must never delete the “base” configuration.

You can modify the:

❑ Console attributes of a configuration.

❑ Configuration properties.

Procedure: How to Modify the Console Attributes of a Configuration

To modify the console attributes of a configuration:

1. In the top pane, click *Management* .
2. Click *Servers* in the Server Management section.
3. In the Configurations section, click the name of the configuration.

   An Update Configuration pane for the selected configuration opens.

4. Change values of console attributes as required and then, click *Finish*.

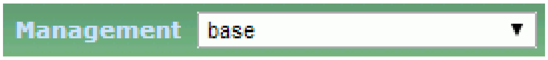
   You are returned to the Configurations pane.

Procedure: How to Modify Configuration Properties

To modify a configuration property:

1. From the Management drop-down list in the top pane of the console home window, select a configuration.

   The following image shows the Management drop-down list.

   Management base ▼

   The middle pane shows the General Properties of the selected configuration.

2. From the main menu, select an option and edit the properties as required.

   The changes apply to the selected configuration.

Stopping a Service

If you add a configuration to iWay Service Manager and later choose to delete it, you must:

1. Stop the service you created for this configuration.
2. Delete the service.
3. Delete the configuration from the console.

Note: Uninstalling iWay Service Manager removes only the configuration and services installed.

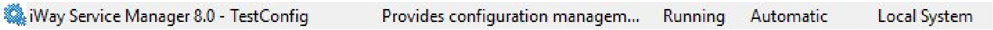
Procedure: How to Stop a Service on Windows

To stop a service on Windows:

1. From the Start menu, select *Start*, *Settings*, *Control Panel*, *Administrative Tools*, and then *Services*.

   The Services window opens.

2. Scroll down to display the iWay Service Manager services, as shown in the following image.

iWay Service Manager 8.0 - TestConfig	Provides configuration managem...	Running	Automatic	Local System

3. Right-click the service you created, for example, iWay Service Manager - TestConfig, and then select *Stop*.

   The service is stopped.

Your next task is to delete the Windows service. For more information, see *How to Remove a Windows Service* on page 101.

Procedure: How to Stop a Service on a non-Windows Platform

To stop a service on a platform other than Windows:

1. Locate the default shutdown file, which is supplied for the base configuration, in the iWay8/bin directory.

   On UNIX, this file is called:

   ```
   stopservice.sh
   ```

   On other non-Windows platforms, the name may vary.

2. Create a copy of this shutdown file for your service.

3. Edit the copy for your service in an editor and navigate to the following section in the file to replace *base* with the name of your configuration:

   ```
   #!/bin/sh
   #####################################################################
   #
   # Init
   # Setup global variables and signal handling
   #
   # Edit following lines to point to install dir, user ID, and config"
   IWAY8=/WorkSource/iWay8/
   IWAYUSER=root
   IWAYCONFIG=base
   ```

4. Navigate to the following section in the file and change the log file name, *serviceShutdown.txt*, to be unique to this service:

   ```
   cd $IWAY8/config/$IWAYCONFIG
   if test 'uname' = 'OS400'; then
       java $REMDBG -cp $CLASSPATH -DIWAY8=$IWAY8
   com.ibi.service.edaqmServiceShutdown -c $IWAYCONFIG >>
   $IWAY8/serviceShutdown.txt &
   elif test 'uname' = 'OS/390'; then    java $REMDBG -cp $CLASSPATH -
   DIWAY8=$IWAY8
   com.ibi.service.edaqmServiceShutdown -c $IWAYCONFIG >>
   $IWAY8/serviceShutdown.txt &
   else    su $IWAYUSER -c "java $REMDBG -cp $CLASSPATH -DIWAY8=$IWAY8
   com.ibi.service.edaqmServiceShutdown -c $IWAYCONFIG >>
   $IWAY8/serviceShutdown.txt &"
   fi
   ```

5. Save your changes.

6. Execute the shutdown file for your service.

   The service is stopped.

Your next task is to delete the service.

Removing a Service and Configuration

This section describes how to remove a service and a configuration.

Procedure: How to Remove a Windows Service

To remove a service that you created to start a configuration:

1. Open a Command Prompt window and navigate to the iWay home bin directory, for example, on Windows, if iWay is installed in C:\Program Files\iWay7, go to:

   ```
   C:\Program Files\iWay7\bin
   ```

2. At the command prompt, type:

   ```
   iwsrv config_name -s remove
   ```

   where:

   config_name

   Is the name of the configuration for which you are deleting a service.

   A message appears, indicating that the service was deleted successfully.

   Your next task is to delete the configuration from the console.

Procedure: How to Remove a Configuration From the Console

To remove a configuration from the console:

1. In the top pane, click *Managed Servers*.

   The Server Management pane opens.

2. Locate the configuration in the Configurations section and select its check box to the left.
3. Click *Delete*.

   A confirmation box opens.

4. Click *OK*.

   The configuration is deleted.

Adding a User

User management in iWay Service Manager (iSM) applies only to users that are authenticated by the default Console Authentication Realm. You can add or delete users, and also assign predefined roles to existing users. More sophisticated and secure user management and role assignment is provided by other authentication realms, for example, using Lightweight Directory Access Protocol (LDAP). LDAP-based realms store user information in a directory structure. Managing users in these realms must be done using the tools provided for these external systems.

The remainder of this section discusses user management only for the default Console Authentication Realm. The other realms that can be used are specified on the Console Settings page of the iSM Administration Console.

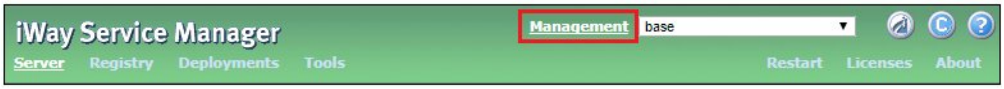
A new (non-administrative) user is automatically granted read and monitor permission for all existing configurations. An administrator can assign restart, stop, create, and delete rights for any user to any or all configurations.

Only an administrator (Power User) can create and delete user accounts. There is no concept of administrator hierarchy; therefore, a newly created administrator has the right to delete another user account, even if that user is an administrator. The system protects the last administrator from being deleted.

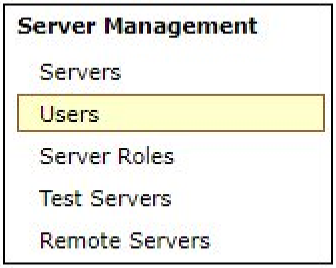
In a production environment, exercise caution in assigning user permissions. For example, it might be reasonable to allow developers to create configurations, but restrict them from deleting them, because the configurations they create might be used in production. Similarly, it might be appropriate to give some users permission to restart configurations, but not to stop them.

Procedure: How to Add a Console Realm User to iWay Service Manager

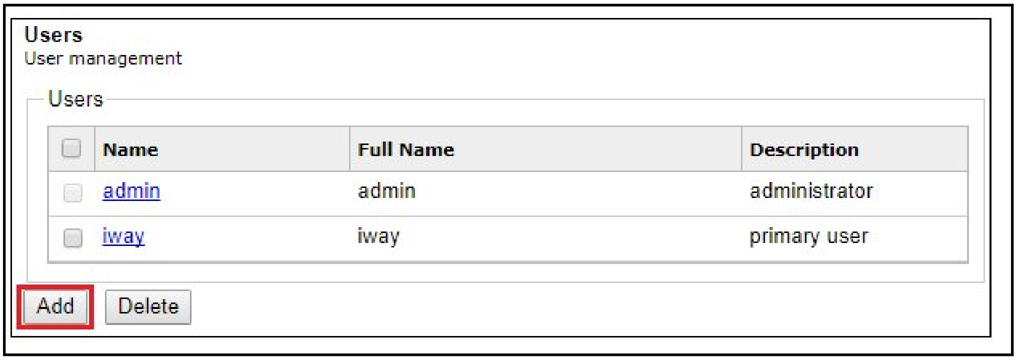
1. In the top pane, click *Management*, as shown in the following image.



2. Click *Users* in the Server Management area, as shown in the following image.



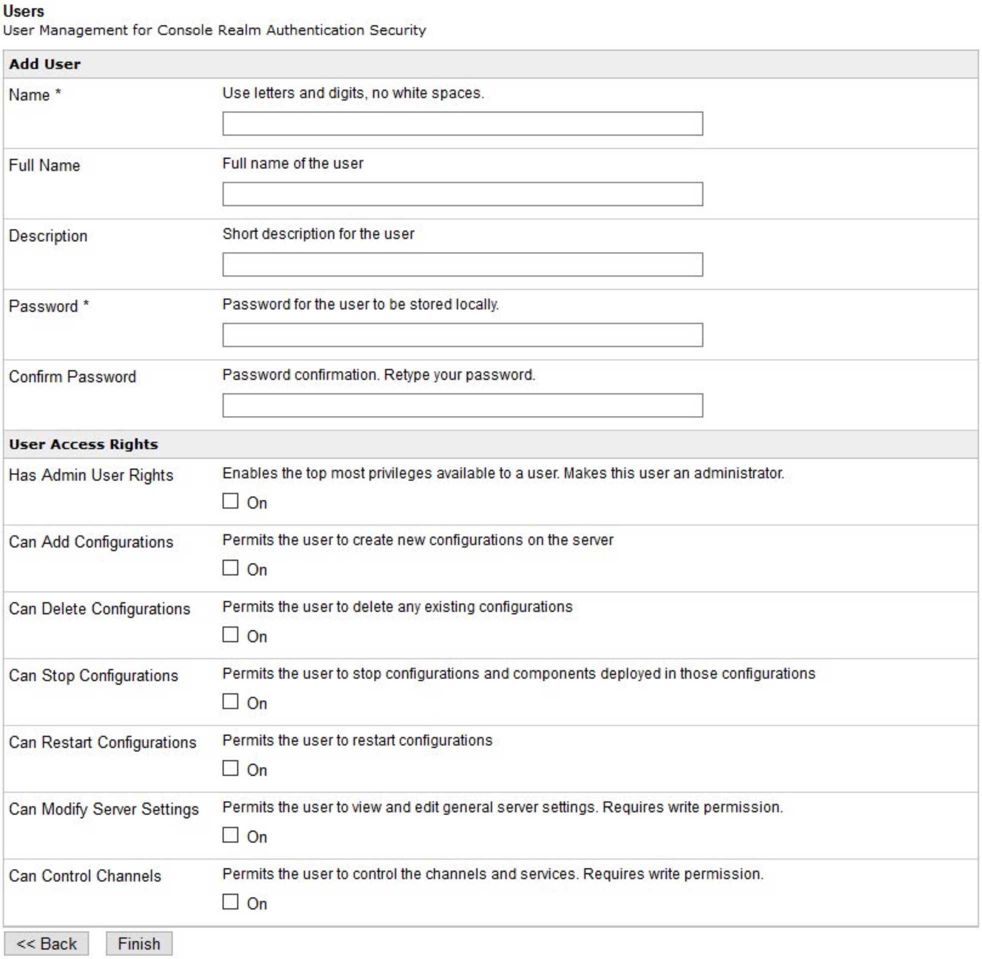
The Users pane opens and lists all of the available users, as shown in the following image.



□	Name	Full Name	Description
□	admin	admin	administrator
□	iway	iway	primary user

3. Click *Add*.

The Add User pane opens, as shown in the following image.



Users

User Management for Console Realm Authentication Security

Add User	
Name *	Use letters and digits, no white spaces.
Full Name	Full name of the user
Description	Short description for the user
Password *	Password for the user to be stored locally.
Confirm Password	Password confirmation. Retype your password.
User Access Rights	
Has Admin User Rights	Enables the top most privileges available to a user. Makes this user an administrator. ☐ On
Can Add Configurations	Permits the user to create new configurations on the server ☐ On
Can Delete Configurations	Permits the user to delete any existing configurations ☐ On
Can Stop Configurations	Permits the user to stop configurations and components deployed in those configurations ☐ On
Can Restart Configurations	Permits the user to restart configurations ☐ On
Can Modify Server Settings	Permits the user to view and edit general server settings. Requires write permission. ☐ On
Can Control Channels	Permits the user to control the channels and services. Requires write permission. ☐ On

a. In the *Name* field, type a short user name (required).

   This value will be the logon ID of the new user, and must not contain any blanks or special characters.

b. In the *Full Name* field, type the full name of the user (optional).

c. In the *Description* field, type a short description (optional).

d. In the *Password* field, type a password.

   The password is stored locally in a masked form.

e. In the *Confirm Password* field, confirm the password by retyping it.

f. Select *display in clear text* to show the password in readable characters (instead of asterisks for security purposes).

4. To enable the required access privileges for the user, select the following check boxes:

   - ❑ Has Admin User Rights (can add, delete, stop, and restart configurations)
   - ❑ Can Add Configurations
   - ❑ Can Delete Configurations
   - ❑ Can Stop Configurations
   - ❑ Can Restart Configurations
   - ❑ Can Modify Server Settings
   - ❑ Can Control Channels

5. Click *Finish*.

   The new user is now added to the list of available users in the Users pane.

Procedure: How to Delete a User From iWay Service Manager

To delete a user from iWay Service Manager:

1. In the top pane, click *Management*.
2. Click *Users* in the Server Management section

   The User Management pane opens.

3. Locate the user in the Users section and select its check box to the left.
4. Click *Delete*.

   A confirmation box opens.

5. Click *OK*.

   The user is deleted.

Managing Configurations in Batch

A batch/shell file (cnfgmgr.bat), available in the iWay7\bin directory, enables you to manage configurations in batch, schedule changes, and integrate with your change management system.

You can:

- ❏ Analyze a configuration. For more information, see *Configuration Analysis Options* on page 106.
- ❏ Create and remove a configuration. For more information, see *Configuration Creation and Removal Options* on page 106 and *Optional Configuration Creation Parameters* on page 107.
- ❏ Update a configuration. For more information, see *Configuration Updating Options* on page 107.
- ❏ Add or delete a package. For more information, see *Options for Adding or Deleting a Package* on page 108.

Reference: Configuration Analysis Options

The following table lists the options for analyzing configurations and their associated commands.

Option	Command
List all configurations.	-listc
List all users for this installation.	-listu
List all packages available for merge.	-listp

Reference: Configuration Creation and Removal Options

The following table lists the options for creating and removing configurations and their associated commands.

Option	Command
Create a new configuration using the base template and the default user name of iway with the password of iway.	-c *name*

Option	Command
Create a new configuration using the default user name of iway with the password of iway, specified template.	-c *name* -template *name*
Create a new configuration based on specified user, base template.	-c *name* -user *name*
Create a new configuration using the specified template and user.	-c *name* -template *name* -user *name*
Create a new configuration based on a dictionary with a full path.	-c *name* -file *name*
Remove the named configuration.	-c *name* -remcfg

Reference: Optional Configuration Creation Parameters

The following table lists the optional parameters available when creating configurations and their associated commands.

Option	Command
Include a description with the configuration.	-description *text*
Specify a port.	-port *number*
Configuration is used as a template only.	-hidden

Reference: Configuration Updating Options

The following table lists the options for updating configurations and their associated commands.

Option	Command
Update console (bport), soap (sport), or http (hport) number to an existing configuration.	-c *name* -sport *number* -bport *number* -hport *number* These can be applied in any combination.

Reference: Options for Adding or Deleting a Package

The following table lists the options for adding or deleting a package and their associated commands.

Option	Command
List all packages available for merge.	-listp
Add packages to a new/existing configuration. Specify multiple packages in the following format: swift/1997, hipaa/4010.	-c *name* -pkg *name/version*
List packages available for unmerge by configuration.	-c *name* -listpr
Remove packages from an existing configuration. Add multiple packages in the following format: swift/1997, hipaa/4010. **Note:** Use the -replace option to override keep package add conflicts.	-c *name* -rempkg -pkg *name*

Example: Listing Configurations

The following example shows how to list the configurations defined in your iSM environment:

```
confgmgr -listc
```

The resulting output includes console port and configuration description information.

```
IWAY 7.0.0 SM home directory: C:\Program Files\iWay7
USAGE:    CnfgMgr -parm value
EXAMPLE:  CnfgMgr -listc (lists all existing configurations)
EXAMPLE:  CnfgMgr -c base2  (creates a new configuration base2)
DEEP (parseconfig) full path to config file is:
C:\Program Files\iWay7\config\config.xml
NAME            PORT      DESCRIPTION
----            ----      -----------
base            9999
demo            52104     null
ntk             10001     null
raw             81
```

Example: Creating a Configuration

The following example shows how to automate the creation of configurations. This example creates a new configuration called xconfig. The configuration is based on a template called raw. The configuration is associated with a specific user ID, that is, iway.

```
cnfgmgr -c xconfig -template raw -user iway
```

The result is:

```
IWAY 7.0.0 SM home directory: C:\Program Files\iWay7
USAGE:    CnfgMgr -parm value
EXAMPLE:  CnfgMgr -listc (lists all existing configurations)
EXAMPLE:  CnfgMgr -c base2  (creates a new configuration base2)
DEEP (parseconfig) full path to config file is: C:\Program Files
\iWay7\config\config.xml
parameters selected:
config = xconfig
template = raw
package(s) = null
port = null
hidden = false
remove(config) = false
remove(package) = false
user = iway
file = null
install directory =
description = null
DEEP (parseconfig) no current configuration by the name xconfig
DEEP (parseconfig) dictionary is: C:\Program Files\iWay7\config\raw\raw.xml
DEEP (genconfig) template dictionary location is:
C:\Program Files\iWay7\config\raw\raw.xml
DEEP (genconfig) new dictionary location is: C:\Program Files\iWay7\config
\xconfig
DEEP (genconfig) New Configuration written to:
C:\Program Files\iWay7\config\xconfig\xconfig.xml
```

Example: Adding an iWay Package to a Configuration

The following example shows how to add an iWay package for Oracle Applications to the xconfig configuration in order to migrate iWay components from production to test. In this example, the package is available in iway\home\etc\manager\packages.

```
cnfgmgr -c xconfig -pkg OracleApplications/1.0
```

The result is:

```
IWAY 7.0.0 SM home directory: C:\Program Files\iWay7
USAGE:    CnfgMgr -parm value
EXAMPLE:  CnfgMgr -listc (lists all existing configurations)
EXAMPLE:  CnfgMgr -c base2  (creates a new configuration base2)
DEEP (parseconfig) full path to config file is: C:\Program Files
\iWay7\config\config.xml
parameters selected:
config = xconfig
template = base
package(s) = OracleApplications/1.0
port = null
hidden = false
remove(config) = false
remove(package) = false
user = iway
file = null
install directory =
description = null
DEEP (packages) package file directory is C:\Program Files\iWay7\etc\manager
\packages
DEEP (packages) package file is
C:\Program Files\iWay7\etc\manager\packages\OracleApplications-package.zip
DEEP (packages)
C:\Program Files\iWay7\config\xconfig\OracleApplications\OracleApplications-
package.xml
already exists on disk and keep option selected. File was not replaced.
DEEP (packages) C:\Program Files\iWay7\etc\misc\oracle\Concurrent.ora
already exists on
disk and keep option selected. File was not replaced.
DEEP (parseconfig) dictionary is: C:\Program Files\iWay7\config\xconfig
\xconfig.xml
DEEP (packages) package OracleApplications has been added to
C:\Program Files\iWay7\config\xconfig\xconfig.xml
```

Chapter 6

Configuring General Properties Using the Console

This section describes how to use the iWay Service Manager Administration Console to configure general properties, for example, system settings, and Java settings.

In this chapter:

iSM Console Settings

In the left console pane of the Server menu, the Settings group contains links to the following iSM settings you can configure:

❑ General Settings

❑ Console Settings

❑ Java Settings

❑ Register Settings

❑ Trace Settings

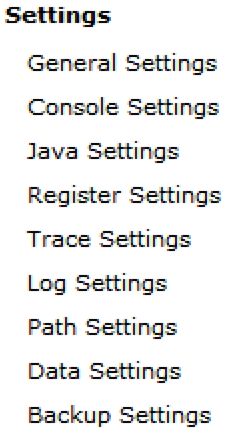
❑ Log Settings

❑ Path Settings

❑ Data Settings

❑ Backup Settings

The following image shows the links available under the Settings group in the iWay Service Manager Administration Console.



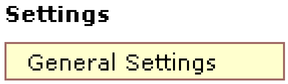
The following sections describes how to configure each of the settings that are available.

General Settings

Using the iWay Service Manager Administration Console, you can configure general settings that are used by the base configuration of the server.

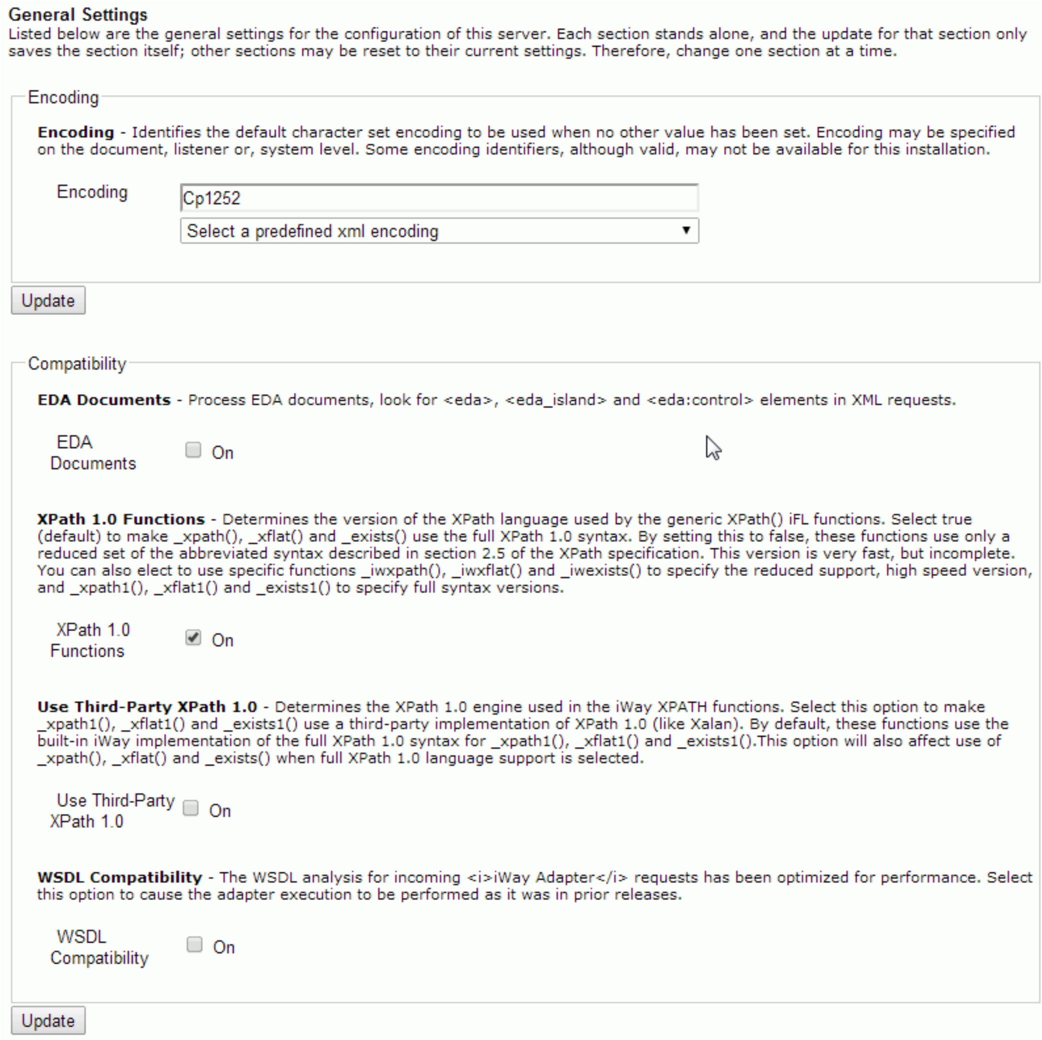
Procedure: How to Configure General Settings

To configure general settings:



1. In the left console pane of the Server menu, select *General Settings*.

The General Settings pane opens, as shown in the following image.



General Settings
Listed below are the general settings for the configuration of this server. Each section stands alone, and the update for that section only saves the section itself; other sections may be reset to their current settings. Therefore, change one section at a time.

Encoding

Encoding - Identifies the default character set encoding to be used when no other value has been set. Encoding may be specified on the document, listener or, system level. Some encoding identifiers, although valid, may not be available for this installation.

Encoding: Cp1252
Select a predefined xml encoding ▾

Update

Compatibility

EDA Documents - Process EDA documents, look for <eda>, <eda_island> and <eda:control> elements in XML requests.

EDA Documents ☐ On

XPath 1.0 Functions - Determines the version of the XPath language used by the generic XPath() iFL functions. Select true (default) to make _xpath(), _xflat() and _exists() use the full XPath 1.0 syntax. By setting this to false, these functions use only a reduced set of the abbreviated syntax described in section 2.5 of the XPath specification. This version is very fast, but incomplete. You can also elect to use specific functions _iwxpath(), _iwxflat() and _iwexists() to specify the reduced support, high speed version, and _xpath1(), _xflat1() and _exists1() to specify full syntax versions.

XPath 1.0 Functions ☑ On

Use Third-Party XPath 1.0 - Determines the XPath 1.0 engine used in the iWay XPATH functions. Select this option to make _xpath1(), _xflat1() and _exists1() use a third-party implementation of XPath 1.0 (like Xalan). By default, these functions use the built-in iWay implementation of the full XPath 1.0 syntax for _xpath1(), _xflat1() and _exists1().This option will also affect use of _xpath(), _xflat() and _exists() when full XPath 1.0 language support is selected.

Use Third-Party XPath 1.0 ☐ On

WSDL Compatibility - The WSDL analysis for incoming <i>iWay Adapter</i> requests has been optimized for performance. Select this option to cause the adapter execution to be performed as it was in prior releases.

WSDL Compatibility ☐ On

Update

2. Type new values or modify the existing values. For more information, see the table in *List of General Settings* on page 113.

3. Click *Update* when you have finished modifying the general settings.

Reference: List of General Settings

The following table lists and describes the available general settings.

Encoding

Property	Type/Value	Description
Encoding	String	Identifies the default character encoding set to use (if not specified in the document or in the listener configuration). Defaults to your local system encoding. You may overwrite the default value with a value from the drop-down list. The values from the drop-down list are: ❑ EUC-JP ❑ ISO-10646-UCS-2 ❑ ISO-10646-UCS-4 ❑ ISO-2022-JP ❑ ISO-8859-1 (US ASCII) ❑ ISO-8859-2 (Eastern Europe) ❑ ISO-8859-3 (Southern Europe) ❑ ISO-8859-4 (Northern Europe) ❑ ISO-8859-5 (ASCII plus Cyrillic) ❑ ISO-8859-6 (ASCII plus Arabic) ❑ ISO-8859-7 (ASCII plus Greek) ❑ ISO-8859-8 (ASCII plus Hebrew) ❑ ISO-8859-9 (Latin 5 Turkish) ❑ ISO-8859-10 (Latin 6 ASCII plus Nordic) ❑ Shift-JIS ❑ UTF-8 ❑ UTF-16 ❑ windows-1252 (Cp1252) - This is the default.

Compatibility

Property	Type/Value	Description
EDA Documents	Check box	Processes EDA documents. Looks for <eda>, <eda_island>, and <eda:control> elements in XML requests.
XPATH 1.0 Functions	Check box	Determines the version of the XPATH language used in some iWay functions. Select this option to make the _xpath() and _exists() functions use the full XPATH 1.0 language. By default, these functions use a simpler version of XPATH to be compatible with prior iSM releases.
Use Third-Party XPath 1.0	Check box	Determines the XPath 1.0 engine used in the iWay XPATH functions. Select this option to make _xpath1(), _xflat1() and _exists1() use a third-party implementation of XPath 1.0 (like Xalan). By default, these functions use the built-in iWay implementation of the full XPath 1.0 syntax for _xpath1(), _xflat1() and _exists1().This option will also affect use of _xpath(), _xflat() and _exists() when full XPath 1.0 language support is selected.
WSDL Compatibility	Check box	The WSDL analysis for incoming <i>iWay Adapter</i> requests has been optimized for performance. Select this option to cause the adapter execution to be performed as it was in prior releases.

Caching

Property	Type/Value	Description
Transform Caching	String	Sets how transforms will be cached in this configuration. Use server level for production runtime. Channel level is best for development in which transforms are modified and republished. Use no caching if transforms will change in a runtime system. This increases latency and should be avoided. Select one of the following options from the drop-down list: ❑ Server: Caching at the server level (default) - This option is selected by default and recommended in production environments. ❑ Channel: Caching at the individual channel level - This option in development environments where transforms are modified and republished. ❑ None: No caching will be performed - Use this option if transforms will change in a runtime system. However, this option also increases latency and should be avoided.

Property	Type/Value	Description
Worker Allocation Strategy	String	How are subchannels (workers) allocated to handle messages? The allocation strategy affects the amount of cache used in the system to retain subchannel resources. This strategy applies to the subset of channel types that directly dispatch work to subschannels (Internal, File, etc). It does not apply to pre-allocated subchannels such as MQ or nHTTP. Under FIFO, subchannels are allocated in order (1, 2, 3...) potentially using more subchannels. Traces are easier to read with this strategy. Under LIFO, subchannels are available for reallocation as they complete processing. This can potentially require fewer subchannels but makes individual subchannel action harder to distinguish in debugging traces. Select from the following options: ❑ FIFO ❑ LIFO

XML DTD Location

Property	Type/Value	Description
Entity DTD Location	String	Location on the file system where DTDs are stored for use. Default is the home directory. As a File-based DTD is accessed during a validating parse, the parser will look in this location for the DTD. ❑ ❑ Type in the location directory. Check the box if a directory does not exist.
Relocate HTTP DTD Entities	Check box	- DTD specified via SystemID in the incoming message can be specified as reached via 'http:' protocol. By checking this option, such DTD's are redirected and located in the Entity DTD Location directory. ❑ Check Redirect *On* to relocate HTTP DTD entities.

Recovery

Property	Type/Value	Description
Configuration Backups	Integer	Number of automatic backups of the configuration (for example, base.xml) to maintain. 0 equals none. If you supply a value greater than 0, the configuration is backed up every time the server starts. It is recommended that you enable this setting.
Configuration Backup Location	Directory	The directory where the configuration backups are saved.
Dead Letter	Directory	Default directory where responses that cannot be delivered are held when the reply-to value cannot be identified. If the directory does not exist, select the check box to create the named directory.
Retry Interval	Duration: *xxhxxmxxs*	Interval (in seconds) after which the listener can be retried if it fails for external causes. The default value is 120. **Note:** The Retry Interval is a global setting that applies to all listeners.
Kill Interval	Duration: *xxhxxmxxs*	Interval at which to check for runaway requests that exceed their maximum life. Default is 60 (seconds), for example, 1h2m3s=1 hour, 2 minutes, and 3 seconds. **Note:** Each listener has an Execution Time Limit property that determines the maximum life of each request.
Startup Process Flow	Process Name	If set, this must be the name of a process flow deployed to the system. The flow will be executed when service manager starts, just prior to the initialization of system exits like activity logs and correlation management. If the process does not complete successfully, service manager will not start. To bypass the startup flow, start the server with the -r switch.

Property	Type/Value	Description
Close Down Process Flow	Process Name	If set, this must be the name of a process flow deployed to the system. The flow will be executed when service manager stops. The closedown will run the configured flow after channels are stopped, but before stopping iSM system services such as connection pools or the scheduler. The result of the flow is ignored, and iSM termination proceeds regardless of how the flow ends. To bypass the closedown flow, start the server with the -r switch.

Security File Location

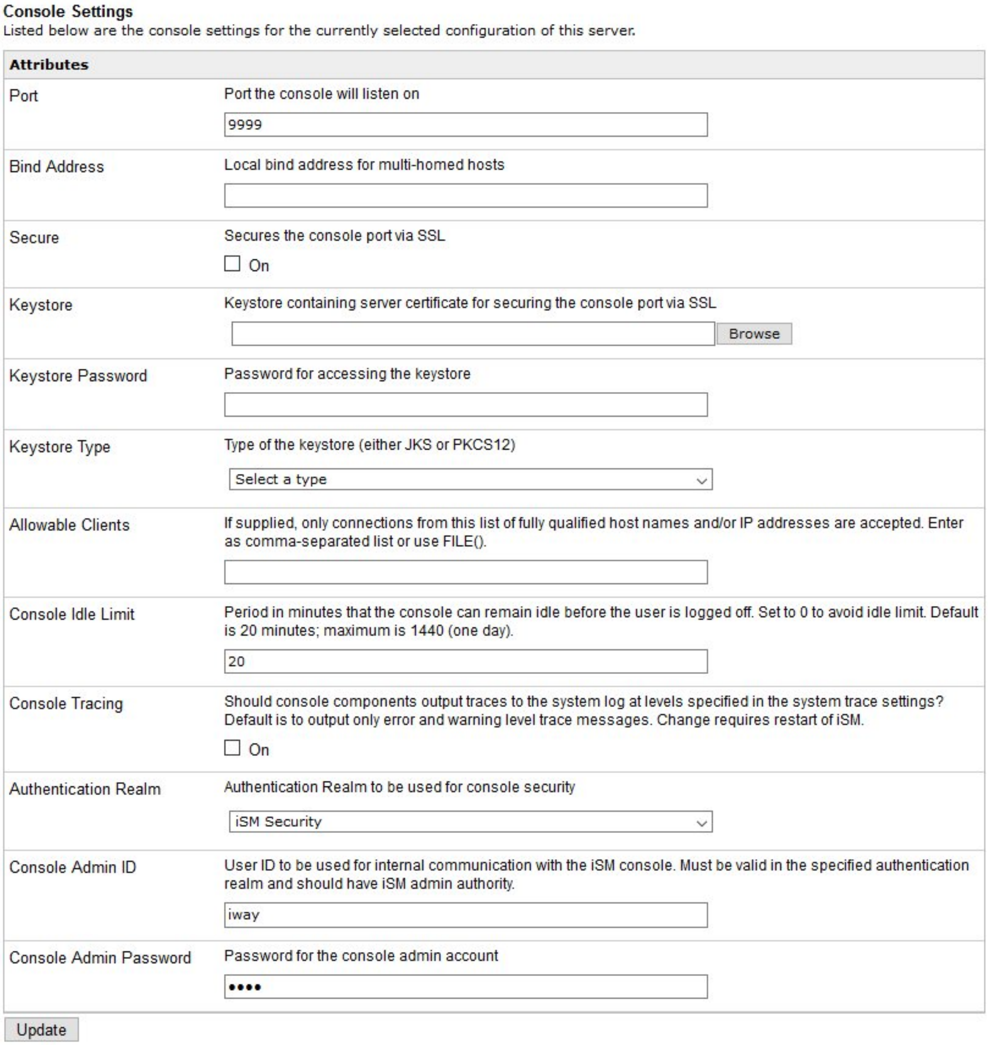
Property	Type/Value	Description
System Security Location	String	The security file location. If may be a file URL or an absolute path location. If omitted, the default is in this installation.

Console Settings

iWay Service Manager (iSM) is configured and monitored using the iWay Service Manager Administration Console, which is a web-based console. The console itself can be configured using the Console Settings pane.

The available settings apply to the master console and are inherited when the console is redirected through the master console configuration (usually base) to other configurations including iWay Integration Applications (iIAs). Only the master configuration supports a console.

The following image shows the Console Settings pane in the iWay Service Manager Administration Console.



Console Settings

Listed below are the console settings for the currently selected configuration of this server.

Attributes	
Port	Port the console will listen on 9999
Bind Address	Local bind address for multi-homed hosts
Secure	Secures the console port via SSL □ On
Keystore	Keystore containing server certificate for securing the console port via SSL Browse
Keystore Password	Password for accessing the keystore
Keystore Type	Type of the keystore (either JKS or PKCS12) Select a type
Allowable Clients	If supplied, only connections from this list of fully qualified host names and/or IP addresses are accepted. Enter as comma-separated list or use FILE().
Console Idle Limit	Period in minutes that the console can remain idle before the user is logged off. Set to 0 to avoid idle limit. Default is 20 minutes; maximum is 1440 (one day). 20
Console Tracing	Should console components output traces to the system log at levels specified in the system trace settings? Default is to output only error and warning level trace messages. Change requires restart of iSM. □ On
Authentication Realm	Authentication Realm to be used for console security iSM Security
Console Admin ID	User ID to be used for internal communication with the iSM console. Must be valid in the specified authentication realm and should have iSM admin authority. iway
Console Admin Password	Password for the console admin account ••••

Update

The following table lists and describes the parameters that are available in the Console Settings pane:

Parameter	Description
Port	The port used to reach the console. The default port is 9999, which is set during the iSM installation.
Bind Address	If the server is multi-homed (connected to more than a single network interface), then this is the address of the interface to be used to reach the console.
Secure	Determines whether to secure the console port using SSL. This option is not enabled by default.
Keystore	If the Secure option is enabled, then this is the path to the keystore holding the private key of the server.
Keystore Password	The password to the keystore.
Keystore Type	The type of keystore being used (JKS or PKCS12).
Allowable Clients	If supplied, only connections from this list of fully qualified host names and/or IP addresses are accepted. Enter as comma-separated list or use FILE().
Console Idle Limit	Approximate time (within 10%) in minutes before an idle console will be logged out by the server.
Console Tracing	Determines whether the iSM Administration Console participates in the standard tracing for the server. If set to *On*, then the console does participate. However, console tracing can be voluminous, and can therefore interfere with trace log analysis. The Console Tracing option is set to *Off* by default, meaning that the console only traces error and warning messages. You can set an individual trace level on the console using the Set command.
Authentication Realm	Choose the Authentication Realm to be used for console security from the drop-down list.
Console Admin ID	The user ID to be used for internal communication with the iSM console.

Parameter	Description
Console Admin Password	The password for the console admin account

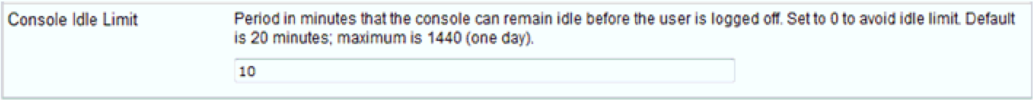
Read-Only User Access

Console protection at the session level is available. Facilities not available to a specific user are disabled, and appear on the iSM Administration Console in the disabled state. Users are provided with the following attributes:

- ❑ **Power user.** As an administrator, the user can use all console facilities.
- ❑ **Modify Server Settings.** Can use the server component of the console.
- ❑ **Control Channels.** Can start and stop channels but cannot change the server configuration. This is usually a system operator.

The console itself requires that the user log on. While it is possible for a single user to be logged on in two or more browser tabs, this is considered a single session by iSM. The user is expected to adhere to good console practices, such as logging off when the console is not in use. A logout link is available on the upper-right corner of the console.

The system administrator can set an idle time (default is 20 minutes) that will be tolerated by the server. If the user does not interact in the session with the server in this period, the user is logged off and must reenter credentials when the console is next used. This option can be set in the Console Settings area of the iSM Administration Console, as shown in the following image.



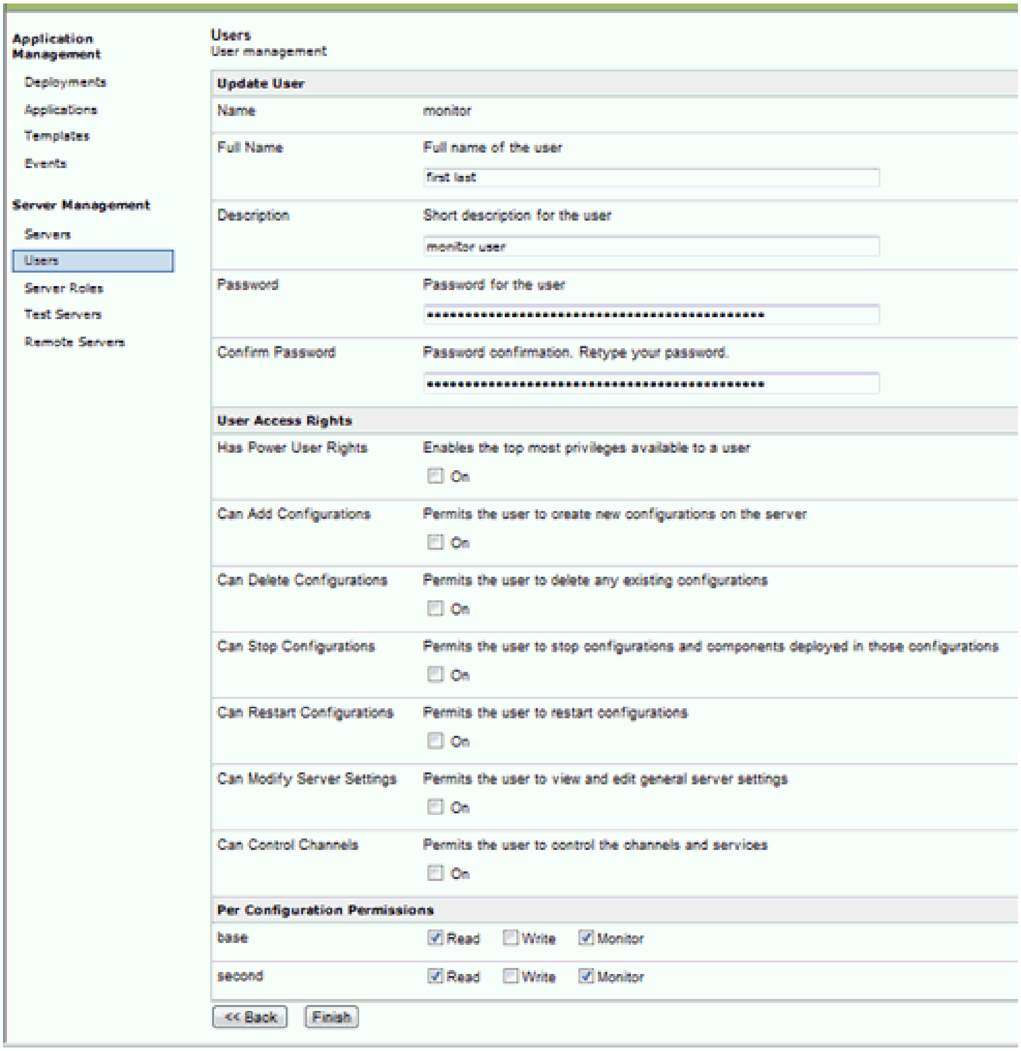
Console Idle Limit

Period in minutes that the console can remain idle before the user is logged off. Set to 0 to avoid idle limit. Default is 20 minutes; maximum is 1440 (one day).

10

In the following example, a user has been created with only monitor rights by not providing any selectable privileges. In this case, when the user logs in to the iSM Administration Console, the user is be able to monitor and view information, but will not be able to modify any iSM settings.

The following image shows the User Management pane in the iSM Administration Console.



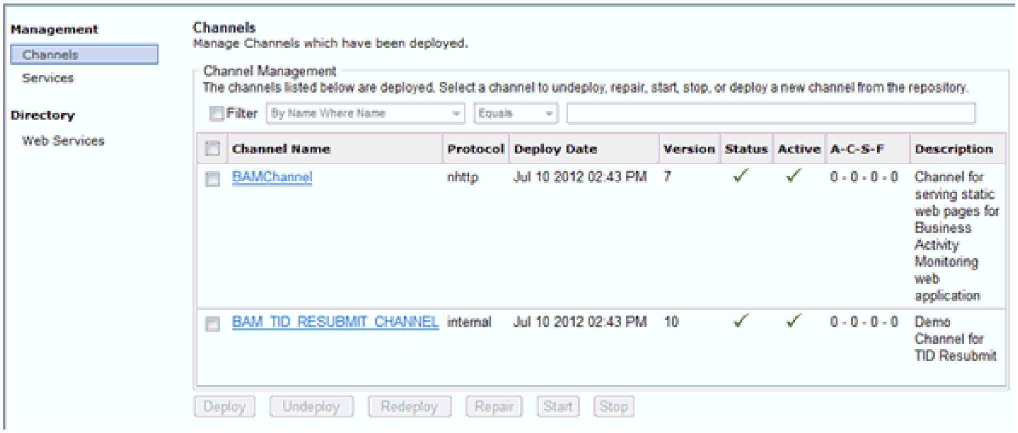
Users
User management

Update User	
Name	monitor
Full Name	Full name of the user first last
Description	Short description for the user monitor user
Password	Password for the user **
Confirm Password	Password confirmation. Retype your password. **

User Access Rights	
Has Power User Rights	Enables the top most privileges available to a user □ On
Can Add Configurations	Permits the user to create new configurations on the server □ On
Can Delete Configurations	Permits the user to delete any existing configurations □ On
Can Stop Configurations	Permits the user to stop configurations and components deployed in those configurations □ On
Can Restart Configurations	Permits the user to restart configurations □ On
Can Modify Server Settings	Permits the user to view and edit general server settings □ On
Can Control Channels	Permits the user to control the channels and services □ On

Per Configuration Permissions	
base	☑ Read □ Write ☑ Monitor
second	☑ Read □ Write ☑ Monitor

The following image shows the Channel Management pane in the iSM Administration Console for a user that has read-only privileges set. Notice the disabled buttons at the bottom of the screen.



Channels
Manage Channels which have been deployed.

Channel Management
The channels listed below are deployed. Select a channel to undeploy, repair, start, stop, or deploy a new channel from the repository.

☐ Filter | By Name Where Name | Equals |

☐	Channel Name	Protocol	Deploy Date	Version	Status	Active	A-C-S-F	Description
☐	BAMChannel	nhttp	Jul 10 2012 02:43 PM	7	✓	✓	0 - 0 - 0 - 0	Channel for serving static web pages for Business Activity Monitoring web application
☐	BAM_TID_RESUBMIT_CHANNEL	internal	Jul 10 2012 02:43 PM	10	✓	✓	0 - 0 - 0 - 0	Demo Channel for TID Resubmit

Java Settings

The Java Settings pane in the iWay Service Manager Administration Console enables you to specify Java Virtual Machine (JVM) options and Java system properties. It also enables you to register Java classes by adding .JAR files to the class path.

Specifying Java Virtual Machine Options

The Java Virtual Machine Settings section enables you to specify the options used for starting the JVM. For example, you can set memory allocation options.

Setting JVM options can improve the performance of Service Manager or correct problems. The most common setting adjustments are for the size of the Java heap and stack, which determine memory availability for Java programs and the JVM. If sufficient memory is not available, errors can occur. The heap size affects performance, as it determines how often garbage collection occurs.

Note: After making any changes to the JVM options, you must completely stop and then start your iSM instance for these changes to take affect.

If you encounter performance problems or receive “out of memory” exceptions, you can adjust these sizes. The following are the JVM memory settings most commonly adjusted:

`-Xss`*###*`M`

Sets the Java thread stack size.

`-Xmx###M`

Sets the maximum Java heap size.

`-Xms###M`

Sets the initial Java heap size.

The size is usually set in megabytes, for example:

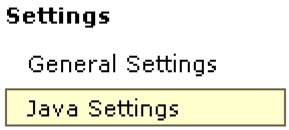
```
-Xmx512M
```

Optimum sizes vary, depending on the total memory available, the requirements of your application, the number of other processes that require memory, the type of JVM, and other considerations.

The location for setting these and other JVM options depends on the operating system. You must edit the UNIX startservice.sh script or the iway7sm.sh script to include JVM options on that platform.

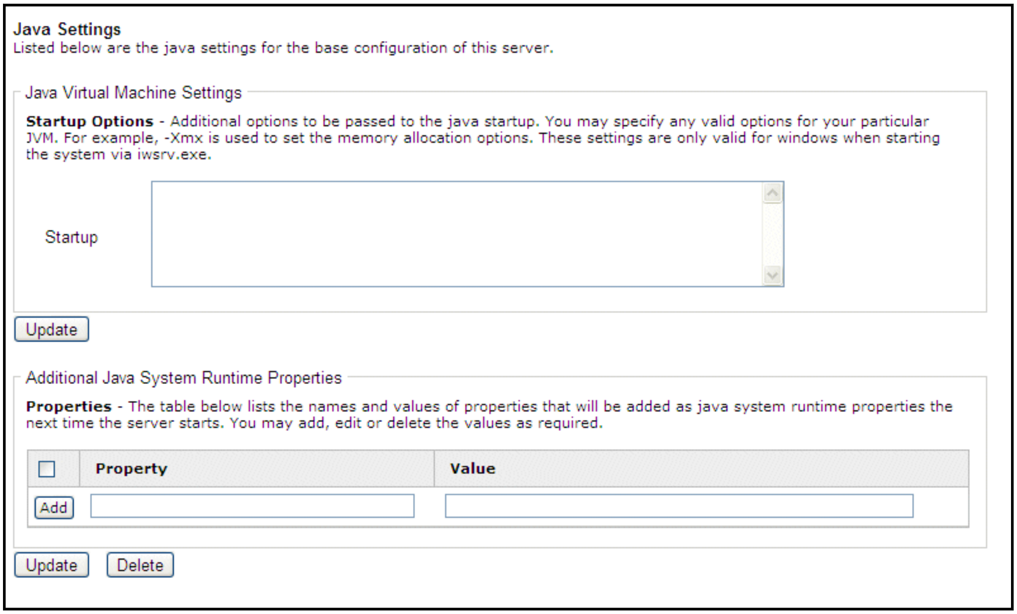
Procedure: How to Specify JVM Options on Windows

To specify JVM options on Windows:



1. In the left console pane of the Server menu, select *Java Settings*.

The Java Settings pane opens, as shown in the following image, displaying two sections: Java Virtual Machine Settings and Additional Java System Runtime Properties.



Java Settings

Listed below are the java settings for the base configuration of this server.

Java Virtual Machine Settings

Startup Options - Additional options to be passed to the java startup. You may specify any valid options for your particular JVM. For example, -Xmx is used to set the memory allocation options. These settings are only valid for windows when starting the system via iwsrv.exe.

Startup

Additional Java System Runtime Properties

Properties - The table below lists the names and values of properties that will be added as java system runtime properties the next time the server starts. You may add, edit or delete the values as required.

□	**Property**	**Value**
Add		

2. In the Java Virtual Machine Settings section, type a JVM memory setting in the Startup field, for example:

   ```
   -Xmx512M
   ```

3. Click *Update*.
4. Due to the way that iSM handles Java class loading, you must completely stop and then start your iSM instance for these changes to take affect.

Procedure: How to Specify JVM Options on a Non-Windows Platform (Service)

To specify JVM options on a non-Windows platform when running Service Manager as a service (daemon):

1. Manually modify the script that starts Service Manager.
2. Edit the script used to start the service (for example, startservice.sh) and add JVM options to the last line, for example:

   ```
   su $IWAYUSER -c "java -Xmx512M
   -Xss256M $REMDBG -cp $CLASSPATH -DIWAY7=$IWAY7sm
   -Diwaysoftware.af.idocument=com.ibi.edaqm.XDDocument
   com.ibi.service.edaqm.XDDocument -$IWAYCONFIG >> $IWAY7sm/bin/
   service.log &"
   ```

Procedure: How to Specify JVM Options on a Non-Windows Platform (Non-Service)

To specify JVM options on a non-Windows platform when running iWay Service Manager manually:

1. Manually modify the script that starts Service Manager.
2. Edit the script used to start Service Manager (for example, iway7sm.sh) and add JVM options to the line that calls the Java command, for example:

   ```
   java -Xmx512M -Xss256M $REMDBG -cp
   $CLASSPATH -DIWAY7=$IWAY7sm edaqm -config $SCRIPT $2 $3 $4 $5 $6
   -Diwaysoftware.af.idocument=com.ibi.edaqm.XDDocument
   ```

Specifying Java System Runtime Properties

The Additional Java System Runtime Properties section enables you to add, edit, or delete Java system properties.

Procedure: How to Specify Java System Runtime Properties

In the Additional Java System Runtime Properties section:

1. In the Property field, type a name for the Java system property.
2. In the Value field, type the value of the Java system property.
3. Click *Add*.
4. Continue adding more system properties as required.
5. Click *Update*.

Register Settings

A Special Register (SREG) is a named variable that referenced a value which is carried throughout the system. Once defined, this variable is available to all components of the system.

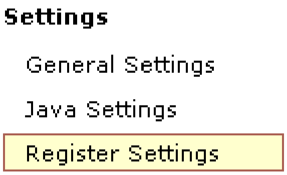
You can examine SREGs as part of the iSM conditional routing facility to control a message flow.

Note: SREGs operate on a global (system-wide) level. As a result, you must restart iSM for any changes to be applied.

For more information on using a SREG to supply configuration properties, see *Configuring Basic Properties* on page 65.

Procedure: How to Define a Special Register (SREG)

To define a Special Register (SREG):



Settings

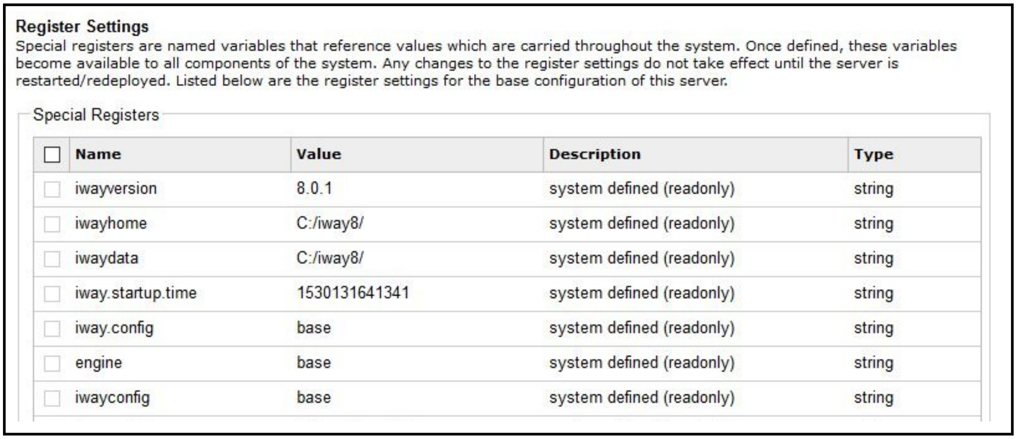
General Settings

Java Settings

Register Settings

1. In the left console pane of the Server menu, select *Register Settings*.

   The Special Registers pane opens, listing the register settings for the base server, as shown in the following image.



Register Settings

Special registers are named variables that reference values which are carried throughout the system. Once defined, these variables become available to all components of the system. Any changes to the register settings do not take effect until the server is restarted/redeployed. Listed below are the register settings for the base configuration of this server.

Special Registers

□	Name	Value	Description	Type
□	iwayversion	8.0.1	system defined (readonly)	string
□	iwayhome	C:/iway8/	system defined (readonly)	string
□	iwaydata	C:/iway8/	system defined (readonly)	string
□	iway.startup.time	1530131641341	system defined (readonly)	string
□	iway.config	base	system defined (readonly)	string
□	engine	base	system defined (readonly)	string
□	iwayconfig	base	system defined (readonly)	string

2. Click *Add*.

   a. In the Name field, type a name for the special register.

   b. From the Type drop-down list, select one of the following:

      ❑ Boolean

      ❑ Duration

      ❑ Float

      ❑ Integer

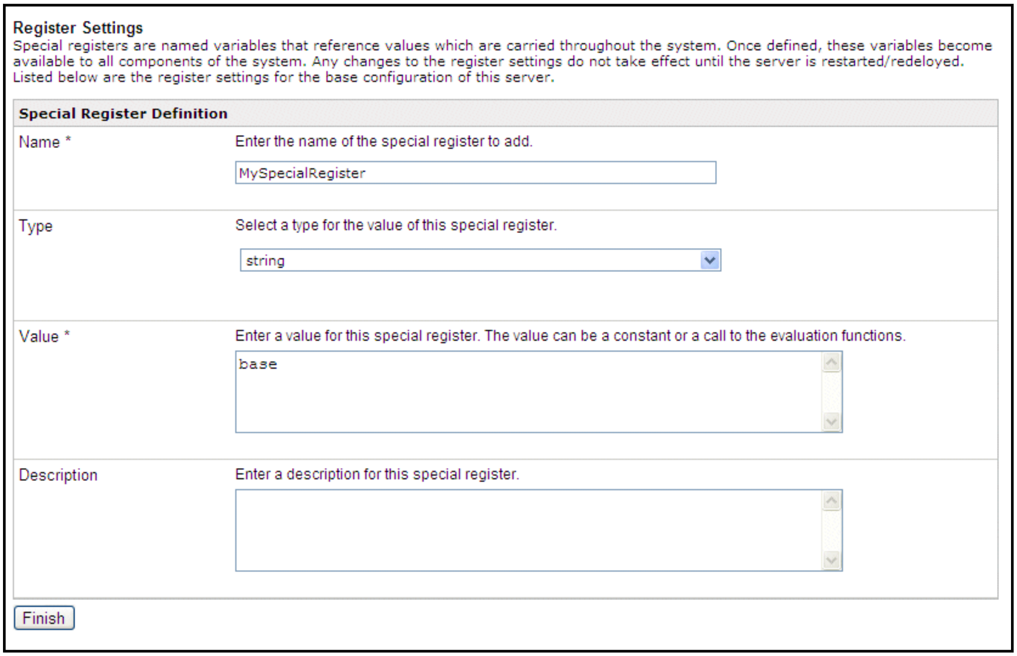
      ❑ Password

      ❑ String

   c. In the Value field, type a value for the special register.

d. In the Description field, type a brief description (optional).

As shown in the following image, the completed Special Register Definition pane shows the Name, Type and Value fields completed.



Register Settings

Special registers are named variables that reference values which are carried throughout the system. Once defined, these variables become available to all components of the system. Any changes to the register settings do not take effect until the server is restarted/redeloyed. Listed below are the register settings for the base configuration of this server.

Special Register Definition

Name *	Enter the name of the special register to add. MySpecialRegister
Type	Select a type for the value of this special register. string
Value *	Enter a value for this special register. The value can be a constant or a call to the evaluation functions. base
Description	Enter a description for this special register.

3. Click *Finish*.

Note: You must completely stop and then start your iSM instance for these changes to take affect.

You are returned to the first Special Registers pane which displayed the newly defined special register. If required, you can now associate the defined special register with a channel. For more information, see *Adding Register Sets* on page 345.

Trace Settings

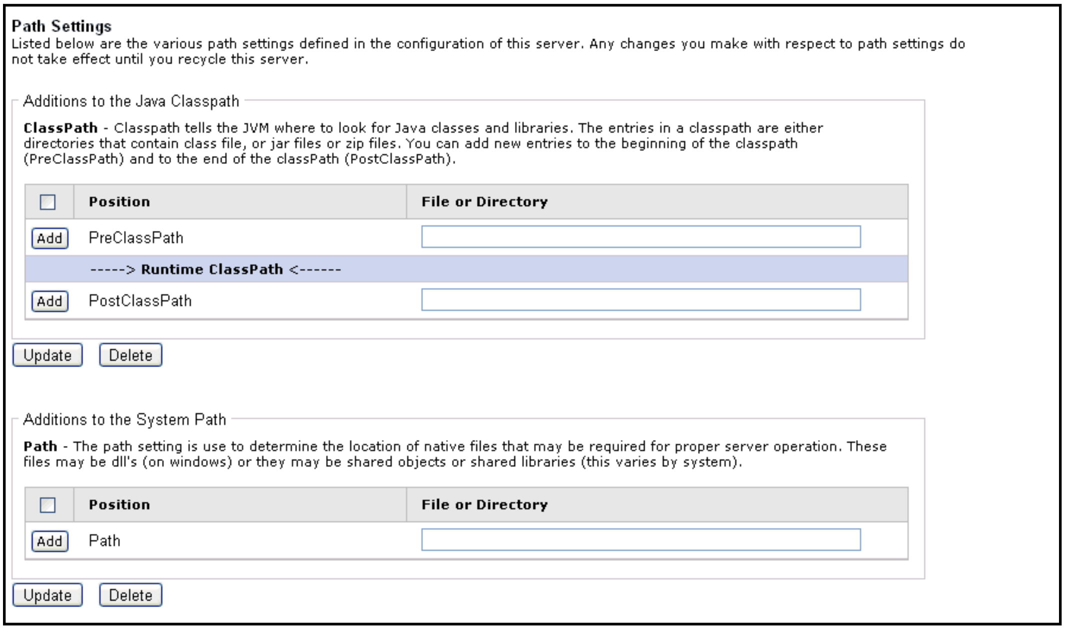
Trace settings allow you to control the amount of detail that is produced by the diagnostic components embedded within iWay Service Manager. Traces produced during run time are either displayed or logged based on settings in the run time environment. For more information on configuring trace settings, see *Diagnostics, Tracing, and Logging* on page 471.

Log Settings

Log settings are used to record the diagnostic information that is generated by the run time components of iWay Service Manager. For more information on configuring log settings, see *Diagnostics, Tracing, and Logging* on page 471.

Path Settings

The Path Settings pane contains the various path settings defined in the configuration of iWay Service Manager.



Path Settings

Listed below are the various path settings defined in the configuration of this server. Any changes you make with respect to path settings do not take effect until you recycle this server.

Additions to the Java Classpath

ClassPath - Classpath tells the JVM where to look for Java classes and libraries. The entries in a classpath are either directories that contain class file, or jar files or zip files. You can add new entries to the beginning of the classpath (PreClassPath) and to the end of the classPath (PostClassPath).

□	**Position**	**File or Directory**
Add	PreClassPath	
	-----> Runtime ClassPath <------	
Add	PostClassPath	

Additions to the System Path

Path - The path setting is use to determine the location of native files that may be required for proper server operation. These files may be dll's (on windows) or they may be shared objects or shared libraries (this varies by system).

□	**Position**	**File or Directory**
Add	Path	

Adding entries to the Java classpath tells the JVM where to look for Java classes and libraries. The entries in a classpath are either directories that contain class files, .jar files, or .zip files. You can add new entries to the beginning of the classpath (PreClassPath) and to the end of the classPath (PostClassPath).

Adding entries to the system path is used to determine the location of native files that may be required for proper server operation. These files can include Dynamic Link Libraries (.dll) on Windows or they may be shared objects or shared libraries, which vary by system.

Important: You must completely stop and then start your iSM instance for any path setting changes to take affect.

Registering Libraries

The Path Settings pane enables you to include JAVA class and JAR files in the class path. Use this function to add third-party drivers, such as those for IBM WebSphere MQ, Oracle AQ, and JDBC. You also can specify additional library directories that may be required when the third-party Java classes require dynamic link libraries or shared objects (depending on the platform in use). WebSphere MQ, for example, requires this type of setting.

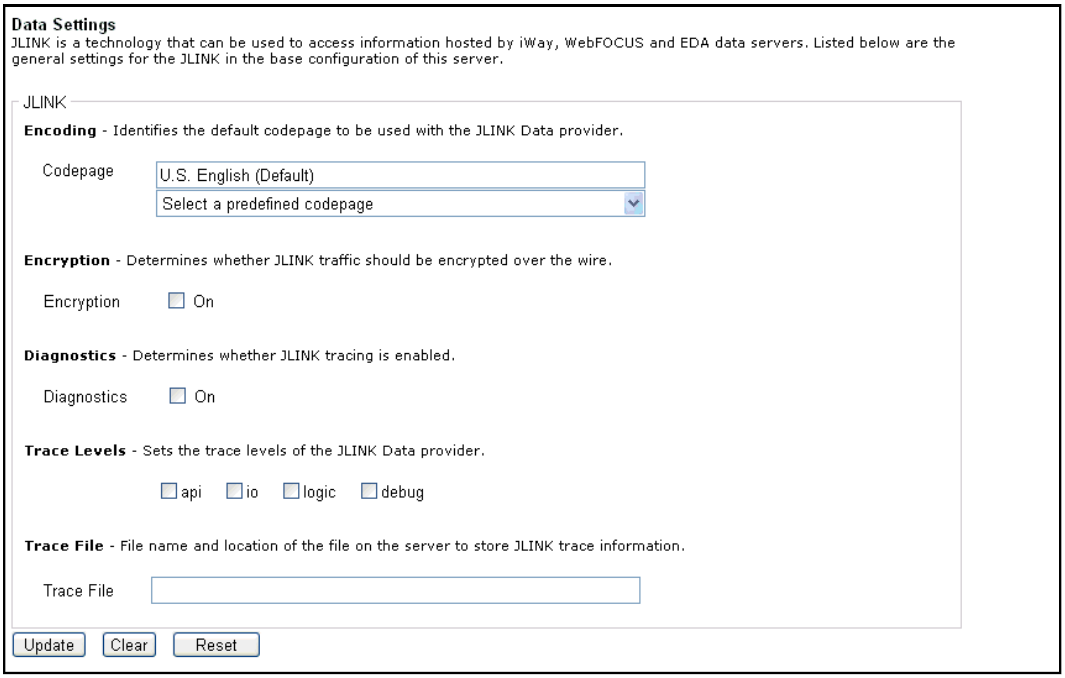
Note: You must completely stop and then start your iSM instance for these changes to take affect. The Restart option on the top navigation pane of the console is not sufficient to implement this type of change.

The following procedure describes how to add third-party Java libraries using IBM WebSphere MQ as an example. First, the third-party JAVA files (JAR, CLASS, or ZIP) must be installed.

Data Settings

JLINK is a technology that can be used to access information hosted by iWay, WebFOCUS, and EDA data servers. The Data Settings pane contains the general settings for the JLINK in the base configuration of this server.

Note: The Data Settings pane is retained for legacy users. It is recommended that all new configurations to data servers should be made through the Data Provider facility in the iWay Service Manager Administration Console. For more information on configuring data providers, see *Configuring a Data Provider*.



Data Settings
JLINK is a technology that can be used to access information hosted by iWay, WebFOCUS and EDA data servers. Listed below are the general settings for the JLINK in the base configuration of this server.

JLINK

Encoding - Identifies the default codepage to be used with the JLINK Data provider.

Codepage: U.S. English (Default)
Select a predefined codepage

Encryption - Determines whether JLINK traffic should be encrypted over the wire.

Encryption □ On

Diagnostics - Determines whether JLINK tracing is enabled.

Diagnostics □ On

Trace Levels - Sets the trace levels of the JLINK Data provider.

□ api □ io □ logic □ debug

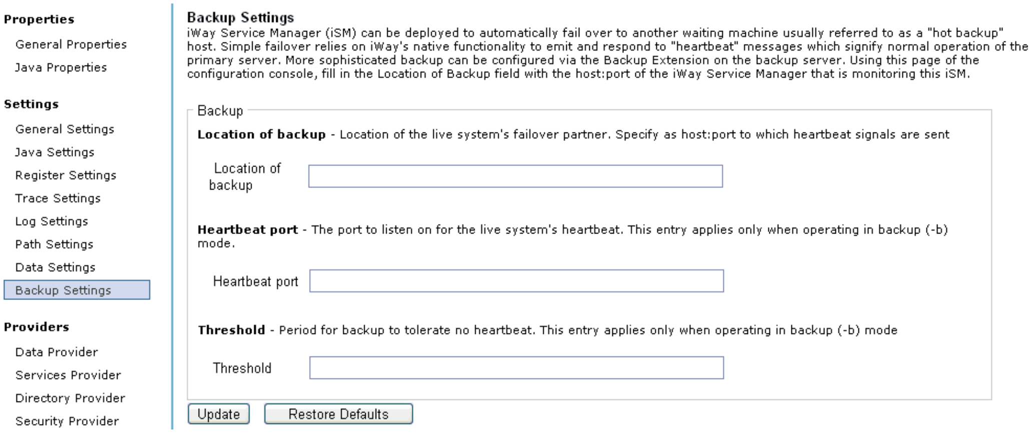
Trace File - File name and location of the file on the server to store JLINK trace information.

Trace File

For more information on configuring the JLINK properties, see *Diagnostics, Tracing, and Logging* on page 471.

Backup Settings

iWay Service Manager (iSM) can be deployed to automatically fail over to another waiting machine usually referred to as a hot backup host. Simple fail over relies on native functionality in iWay to emit and respond to "heartbeat" messages, which signify normal operation of the primary server. More sophisticated backup can be configured via the Hot Backup extension on the backup server. In the Backup Settings pane of the iWay Service Manager Administration Console, provide a hostname and port number in the Location of Backup field that corresponds to the iWay Service Manager that is monitoring this instance of iSM.



Backup Settings

iWay Service Manager (iSM) can be deployed to automatically fail over to another waiting machine usually referred to as a "hot backup" host. Simple failover relies on iWay's native functionality to emit and respond to "heartbeat" messages which signify normal operation of the primary server. More sophisticated backup can be configured via the Backup Extension on the backup server. Using this page of the configuration console, fill in the Location of Backup field with the host:port of the iWay Service Manager that is monitoring this iSM.

Backup

Location of backup - Location of the live system's failover partner. Specify as host:port to which heartbeat signals are sent

Location of backup

Heartbeat port - The port to listen on for the live system's heartbeat. This entry applies only when operating in backup (-b) mode.

Heartbeat port

Threshold - Period for backup to tolerate no heartbeat. This entry applies only when operating in backup (-b) mode

Threshold

For more information about configuring the Hot Backup extension, see the *iWay Service Manager Extensions User's Guide*.

iSM Providers

A provider is a centrally configured resource that supplies services to run time components in the server. For example, a keystore provider centralizes the definition of one security keystore, including its type, file location, and password. Each configured provider has a name. Using that name the services of the provider can be referenced in other parts of the server.

One provider can refer to another provider. The SSL provider, as an example, requires a keystore and a trust store. Each of these is a keystore of some type, so instead of configuring all of the details of the keystore, the configuration simply asks for the name of the keystore provider(s) in charge of the keystore and trust store.

Often, several providers can supply the same service, although in different ways. For example, in a secure system a certificate can be stored in a keystore or in an LDAP directory. A Certificate Revocation List (CRL) can be stored in a file system directory or in an LDAP directory. Simply specifying the name of the provider to be used to access the certificate or the CRL is all that is needed when configuring for a need. This simplifies server configuration.

A provider describes a resource available at run time, while the users of the provider are configured in the design time experience; a deployed usage is tied to the run time physical implementation only by its name. For example, a configuration requiring a certificate store can be deployed on servers having completely different storage for its certificates.

Please note that the term providers is used on several levels. While the providers available in iWay Service Manager offer services to other iWay components, providers can also refer to software that is installed into the Java Virtual Machine (JVM) that provides services to application programs. These JVM providers must also be configured.

As with all server-level modifications, you must completely stop and then start your iSM instance for these changes to take affect.

Data Provider

The Data Provider option enables you to define and configure data servers and connections. The data provider properties include:

- ❏ JDBC connections
- ❏ JLINK to access iWay, WebFOCUS, and EDA data servers

Procedure: How to Add a JDBC Connection

To add a JDBC connection:

1. In the left console pane of the Server menu, select *Data Provider*.
2. Beneath the JDBC section, click *New*.

   The JDBC Data Provider Definition pane displays.

3. Provide the appropriate values for your JDBC connection as listed and defined in the following table.

JDBC Connection Pool Properties

Parameter	Description
Name *	Enter the name of the JDBC data provider to add.

Parameter	Description
Driver Class	The JDBC driver class is the name of the class that contains the code for this JDBC Driver. You can select a predefined database from the drop-down list or enter your own. The following are sample values for the driver: com.microsoft.jdbc.sqlserver.SQLServerDriver COM.ibm.db2.jdbc.app.DB2Driver com.informix.jdbc.IfxDriver sun.jdbc.odbc.JdbcOdbcDriver oracle.jdbc.driver.OracleDriver com.sybase.jdbc2.jdbc.SybDriver The required .jar files for the JDBC driver must be installed and registered in the Service Manager classpath. You can use the Path Settings option on the console to add Java classes and libraries.
Connection URL	The JDBC connection URL to use when creating a connection to the target database. The URL generally includes the server name or IP address, the port or service, the data source name, and a driver specific prefix. You can select a predefined database from the drop-down list or enter your own. The following are sample values for the URL: jdbc:microsoft:sqlserver://*server*:1433;DatabaseName=DB jdbc:db2:*database* jdbc:informix-sqli://*HOST:PORT/* *DB*:INFORMIXSERVER=SERVER_NAME jdbc:odbc:*DB*jdbc:oracle:*thin:@HOST:PORT:SID*jdbc:sybase :Tds:*HOST:PORT* For more information, see the JDBC documentation for the specific data source.
User	User name with respect to the JDBC URL and driver. SREG names can be used.
Password	Password with respect to the JDBC URL and driver. SREG names can be used.

Connection Pool Properties

Parameter	Description
Initial Pool Size *	Number of connections to place in the pool at startup.
Maximum Number of Idle Connections *	Maximum number of idle connections to retain in the pool. A value of zero (0) means no limit except what is enforced by the maximum number of connections in the pool. This value can be reset using the *jdbc* command, allowing this to be changed, often on a schedule, to respond to changing database conditions.
Maximum Number of Connections *	Maximum number of connections in the pool. A value of zero (0) means no limit. This value can be reset using the *jdbc* command, allowing this to be changed, often on a schedule, to respond to changing database conditions.
Login Timeout	Time in seconds to wait for a pooled connection before throwing an exception. A value of zero (0) means to wait forever.
Behavior When Exhausted	What to do when the pool reaches the maximum number of connections. Block means wait for a connection to become available for the period defined by the login timeout parameter. Fail means throw an exception immediately.
Cache Prepares	Determines whether the provider should cache prepared statements. Set this to false if your application generates a large number of prepared or callable statements that are not reused.

Idle Connection Eviction

This feature allows you to specify the amount of time a connection can remain idle in the connection pool. Similar to a feature of Apache DBCP, the idle connection eviction thread runs at regular intervals, with the following options:

Parameter	Description
Eviction Interval	Time between runs of the idle connection eviction thread, in seconds. A negative value means the eviction thread will never run.
Idle Connection Timeout	The minimum amount of time a connection can remain idle in the pool before the eviction thread disposes of it. When the eviction thread runs and finds that a connection has been idle for at least this interval, the connection will be removed from the pool and closed. Note that a connection can remain in the pool longer than the timeout, depending on the scheduling of the eviction interval.
Maximum Number of Tests Per Run	This is a performance tuning option. Since the eviction thread must lock the pool when it runs, this option allows you to specify how many connections can be tested in each run, thus limiting the duration of the lock. If the eviction interval is short, or the number of connections in the pool is large, performance may improve with fewer connections tested per run. Enter 0 to test all connections.

Connection Validation

Parameter	Description
Validation SQL	SQL statement that can be executed to validate the health of a pooled connection. The statement should return a result set of at least one row.
Validate on Borrow	If set to *true*, the validation SQL statement will be executed on a pooled connection before returning the connection to the caller.
Validate on Return	If set to *true*, the validation SQL statement will be executed on a pooled connection before replacing the connection in the pool.

Parameter	Description
Validate Idle	If true, the eviction thread executes the validation SQL statement on connections that have not reached the idle connection timeout. If the statement does not execute successfully, the connection will be dropped from the pool.

4. Click *Test* to check for proper connections.

   If a connection cannot be made, an error message displays describing the problem. Typically, the driver has not been installed or the classpath has not been set.

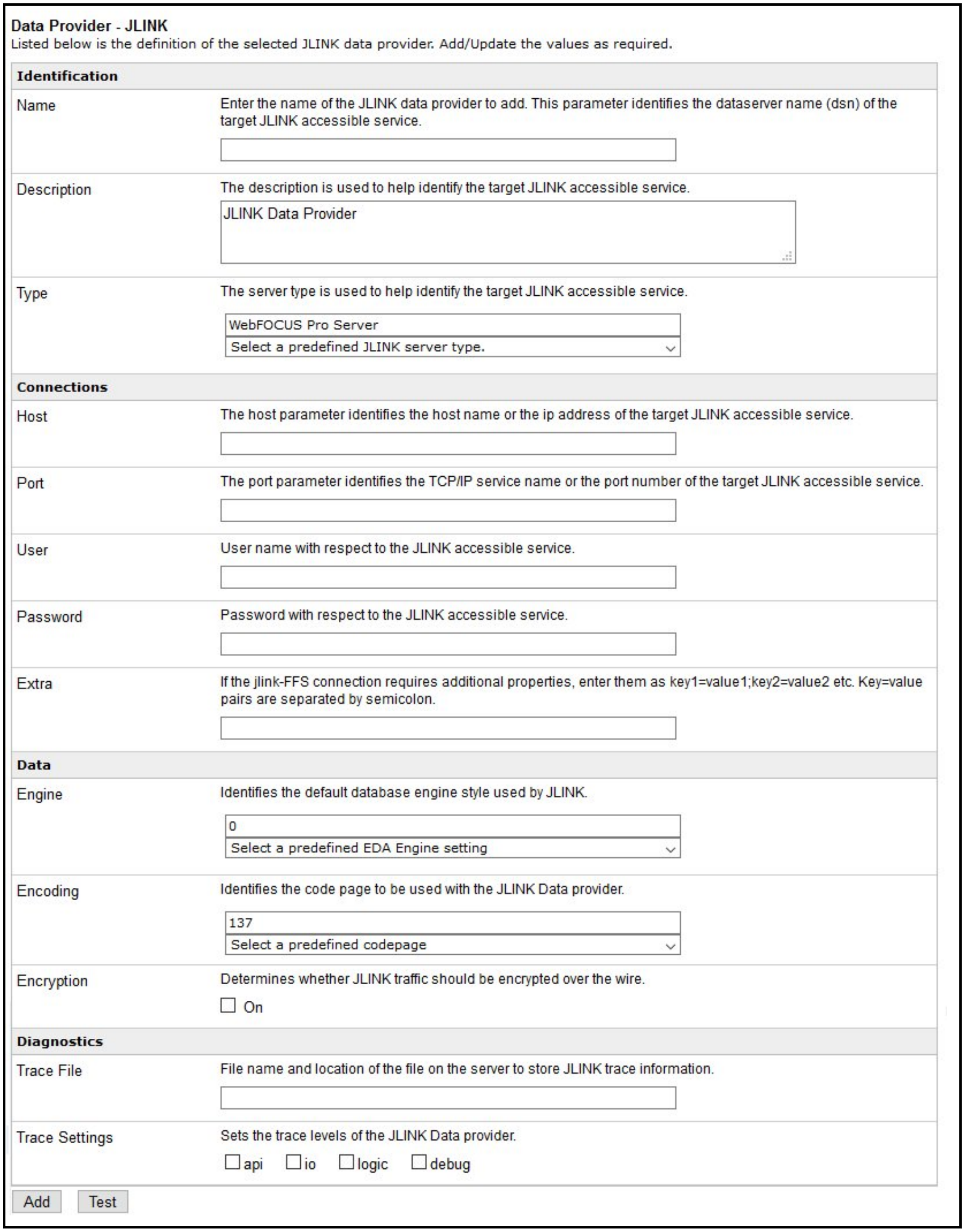
5. Click *Add* to return to the Data Provider pane.

Procedure: How to Add a JLINK Data Source

Since drivers are stopped, they do not need to be added. However, if you need to add a JLINK data source:

1. In the left console pane of the Server menu, select *Data Provider*.
2. Beneath the JLINK section, click *New*.

The Data Provider - JLINK pane displays, as shown in the following image.



Data Provider - JLINK

Listed below is the definition of the selected JLINK data provider. Add/Update the values as required.

Identification

Name: Enter the name of the JLINK data provider to add. This parameter identifies the dataserver name (dsn) of the target JLINK accessible service.

Description: The description is used to help identify the target JLINK accessible service.
JLINK Data Provider

Type: The server type is used to help identify the target JLINK accessible service.
WebFOCUS Pro Server
Select a predefined JLINK server type.

Connections

Host: The host parameter identifies the host name or the ip address of the target JLINK accessible service.

Port: The port parameter identifies the TCP/IP service name or the port number of the target JLINK accessible service.

User: User name with respect to the JLINK accessible service.

Password: Password with respect to the JLINK accessible service.

Extra: If the jlink-FFS connection requires additional properties, enter them as key1=value1;key2=value2 etc. Key=value pairs are separated by semicolon.

Data

Engine: Identifies the default database engine style used by JLINK.
0
Select a predefined EDA Engine setting

Encoding: Identifies the code page to be used with the JLINK Data provider.
137
Select a predefined codepage

Encryption: Determines whether JLINK traffic should be encrypted over the wire.
□ On

Diagnostics

Trace File: File name and location of the file on the server to store JLINK trace information.

Trace Settings: Sets the trace levels of the JLINK Data provider.
□ api □ io □ logic □ debug

3. In the Name field, type the name of a new server. In this example, type *NEWSERV*.

4. In the Description field, type a brief description for the new server. The default is JLINK Data Provider.
5. From the Type drop-down list, select a JLINK server type.

   The default is WebFOCUS Pro Server.
6. Type the required values for Host and Port.
7. Type the required values for User and Password.
8. From the Engine drop-down list, select a database engine.

   The default is 0 (EDA).
9. From the Encoding drop-down list, select a codepage.

   The default value is 137 (U.S. English).
10. To encrypt data that is transported over the wire (optional), select the *Encryption* check box.
11. In the Trace File field, type the path and name of the file for the trace output.
12. To set trace levels, select any number of the check boxes listed (optional).
13. Click *Add*.

Services Provider

iWay Service Manager can quickly and easily expose iWay Business Services as web services through the iWay Business Services Provider (iBSP).

iWay Business Services Provider is installed with an embedded HSQL repository, which is the default data store for information that is generated during design time and then published into a deployed run-time environment. HSQL is an open source Java-based SQL relational database engine and includes a JDBC driver.

iBSP uses defined services providers to integrate with available data repositories. If a repository is not defined, the default HSQL repository is assumed by iWay Service Manager.

A repository migration facility is also included, which provides portability for your existing metadata and iWay Business Services. For example, you can migrate your data between development, testing, and production environments across multiple systems. For more information, see *Migrating Repositorie*s.

The settings in the Services Provider pane refer to the configuration of the iWay Business Services Provider in the base configuration of the server.

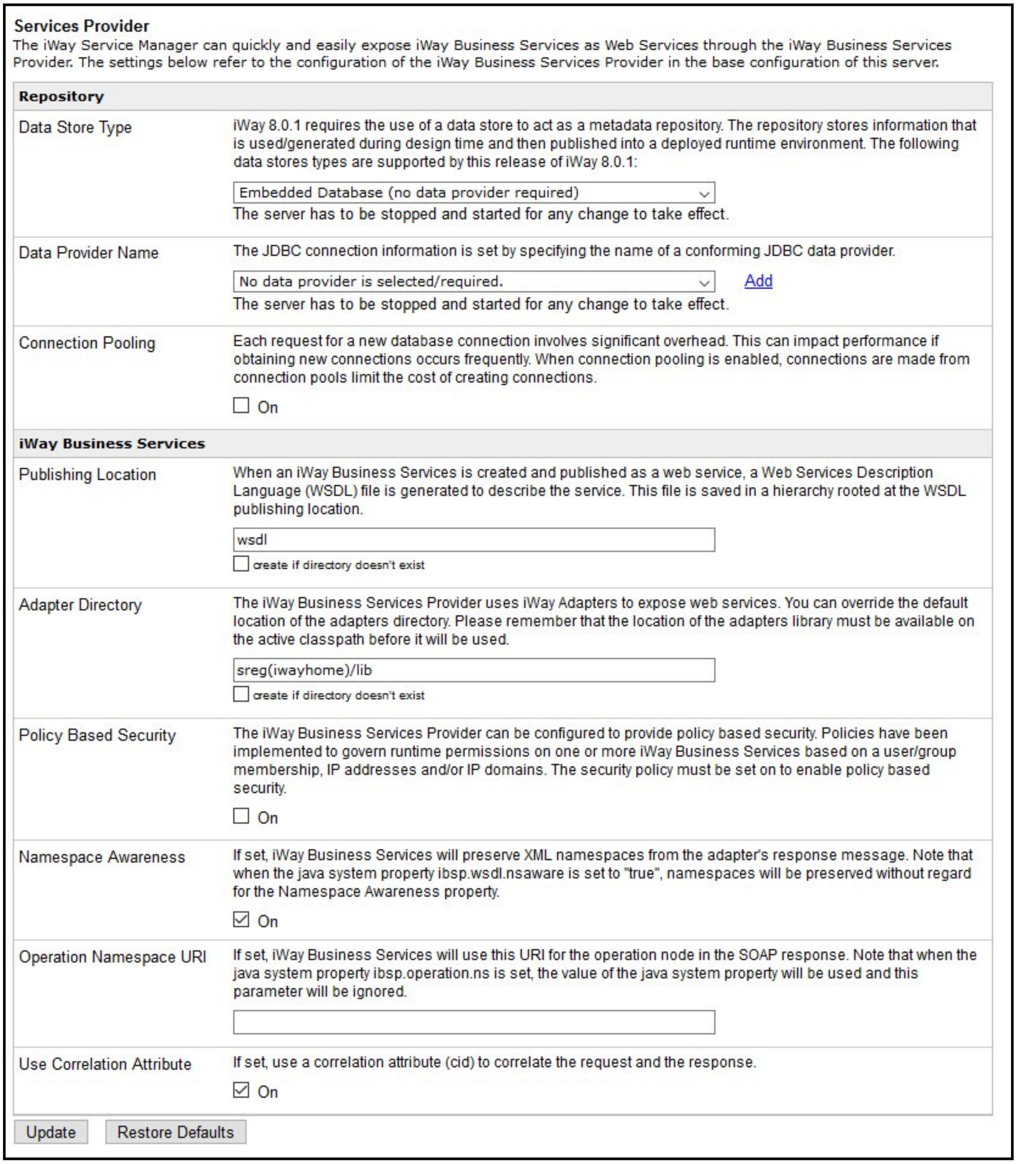
Procedure: How to Configure Services Provider Settings

The Services Provider pane enables you to define properties required to support iWay business services as web services.

To configure for a web service:

1. In the left console pane of the Server menu, select *Services Provider*.

The Services Provider pane opens, as shown in the following image.



Services Provider

The iWay Service Manager can quickly and easily expose iWay Business Services as Web Services through the iWay Business Services Provider. The settings below refer to the configuration of the iWay Business Services Provider in the base configuration of this server.

Repository	
Data Store Type	iWay 8.0.1 requires the use of a data store to act as a metadata repository. The repository stores information that is used/generated during design time and then published into a deployed runtime environment. The following data stores types are supported by this release of iWay 8.0.1: Embedded Database (no data provider required) The server has to be stopped and started for any change to take effect.
Data Provider Name	The JDBC connection information is set by specifying the name of a conforming JDBC data provider. No data provider is selected/required. Add The server has to be stopped and started for any change to take effect.
Connection Pooling	Each request for a new database connection involves significant overhead. This can impact performance if obtaining new connections occurs frequently. When connection pooling is enabled, connections are made from connection pools limit the cost of creating connections. ☐ On
iWay Business Services	
Publishing Location	When an iWay Business Services is created and published as a web service, a Web Services Description Language (WSDL) file is generated to describe the service. This file is saved in a hierarchy rooted at the WSDL publishing location. wsdl ☐ create if directory doesn't exist
Adapter Directory	The iWay Business Services Provider uses iWay Adapters to expose web services. You can override the default location of the adapters directory. Please remember that the location of the adapters library must be available on the active classpath before it will be used. sreg(iwayhome)/lib ☐ create if directory doesn't exist
Policy Based Security	The iWay Business Services Provider can be configured to provide policy based security. Policies have been implemented to govern runtime permissions on one or more iWay Business Services based on a user/group membership, IP addresses and/or IP domains. The security policy must be set on to enable policy based security. ☐ On
Namespace Awareness	If set, iWay Business Services will preserve XML namespaces from the adapter's response message. Note that when the java system property ibsp.wsdl.nsaware is set to "true", namespaces will be preserved without regard for the Namespace Awareness property. ☑ On
Operation Namespace URI	If set, iWay Business Services will use this URI for the operation node in the SOAP response. Note that when the java system property ibsp.operation.ns is set, the value of the java system property will be used and this parameter will be ignored.
Use Correlation Attribute	If set, use a correlation attribute (cid) to correlate the request and the response. ☑ On

2. From the Data Store Type drop-down list, select a repository you want to configure.

   The following are available:

   ❑ Embedded Database (no data provider required)

❏ File System (no data provider required)

❏ IBM DB2

❏ MaxDB

❏ Microsoft SQL Server

❏ Oracle

❏ Sybase

You must configure the tables before using the repository. For more information on configuring repository tables, see the *iWay Installation and Configuration Guide*.

Note: iWay Service Manager is installed with an embedded HSQL repository, which is the default data store for information that is generated during design time and then published into a deployed runtime environment.

For more information on the properties on this window, see the table in *Services Provider Settings*.

3. From the Data Provider Name drop-down list, select an available data provider.

   For more information on how to add a data provider, see *Add a Data Provider*.

4. Type new values or modify existing values.
5. Click *Update*.

Reference: Services Provider Settings

The following table lists and describes the Services Provider settings.

Property	Type/ Value	Description
Repository		
Data Store Type	Choice	Acts as a metadata repository. The default value is Embedded Database. Select a database from the drop-down list: File System (not supported for production use), IBM DB2, MaxDB, Microsoft SQL Server, Oracle, and Sybase.

Property	**Type/ Value**	**Description**
Data Provider Name	String	JDBC driver defined in the Data Provider pane. Select from the drop-down list, or click Add to define a new JDBC connection.
Connection Pooling	Boolean	When selected, turns on connection pooling for the JDBC driver.
iWay Business Services		
Publishing Location	Director y	Directory where the WSDL files produced by the iWay Business Services Provider (iBSP) are stored. If the directory does not exist, select the check box to create the named directory.
Adapter Library	Director y	Directory where the iWay adapter JAR files are located. If the directory does not exist, select the check box to create the named directory.
Policy Based Security	Boolean	When selected, enforces iBSP security policy. For more information, see the *iWay Business Services Provider User's Guide*.
Namespace Awareness	Boolean	If set, iWay Business Services preserve XML namespaces from the adapter's response message in the SOAP response message. **Note:** When the Java system property `ibsp.wsdl.nsaware` is set to *true*, namespaces are preserved without regard for the Namespace Awareness property.
Operation Namespace URI	String	If set, iWay Business Services use this specified URI for the operation node in the SOAP response message. **Note:** When the Java system property `ibsp.operation.ns` is set, the value of this Java system property is used and the Operation Namespace URI property is ignored.
Use Correlation Attribute	Boolean	If set, use a correlation attribute (cid) to correlate the request and the response.

Additional Notes on the Operation Namespace URI Property

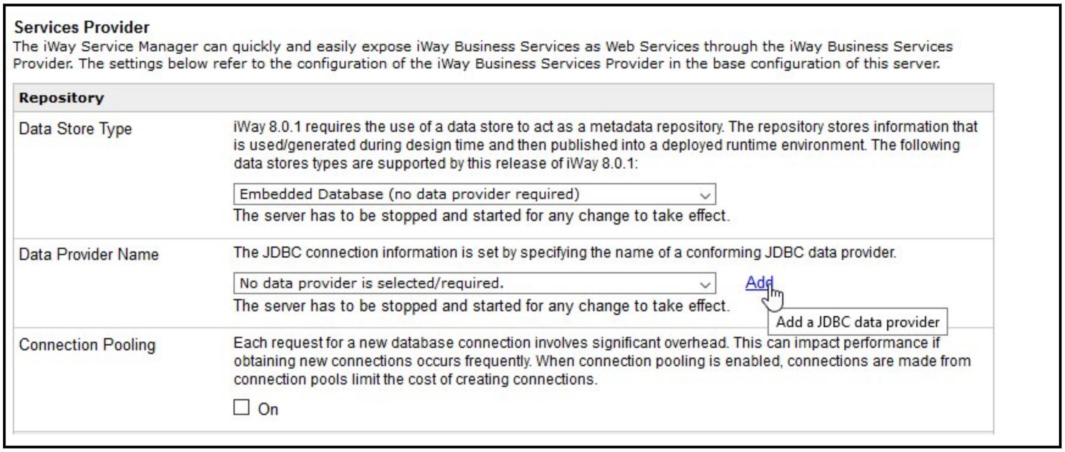
In this example, an iWSE SOAP response from a web service method named PFIVP_1 would have the following format:

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
     <SOAP-ENV:Body>
        <PFIVP_1Response cid="9952CD68D2AFAC3FCD1ED00A645237D4"
         xmlns="urn:iwaysoftware:ibse:jul2003:PFIVP_1:response">
              <d>
                  <d1/>
                  <d2/>
              </d>
        </PFIVP_1Response>
     </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Note that the child of the SOAP Body element is named PFIVP_1Response and it declares a default namespace of urn:iwaysoftware:ibse:jul2003:PFIVP_1:response. This element is the parent of the actual output from the adapter. The element name and the namespace URI are constructed using the name of the service method. In a use case scenario, you may not want the response from an iWSE service to reference iWay Software, as the default namespace URI does. You can use the Operation Namespace URI parameter to override the default behavior by supplying your own namespace URI.

Procedure: How to Add a Data Provider

To add a data provider:



Services Provider

The iWay Service Manager can quickly and easily expose iWay Business Services as Web Services through the iWay Business Services Provider. The settings below refer to the configuration of the iWay Business Services Provider in the base configuration of this server.

Repository

Data Store Type

iWay 8.0.1 requires the use of a data store to act as a metadata repository. The repository stores information that is used/generated during design time and then published into a deployed runtime environment. The following data stores types are supported by this release of iWay 8.0.1:

Embedded Database (no data provider required)

The server has to be stopped and started for any change to take effect.

Data Provider Name

The JDBC connection information is set by specifying the name of a conforming JDBC data provider.

No data provider is selected/required.

The server has to be stopped and started for any change to take effect.

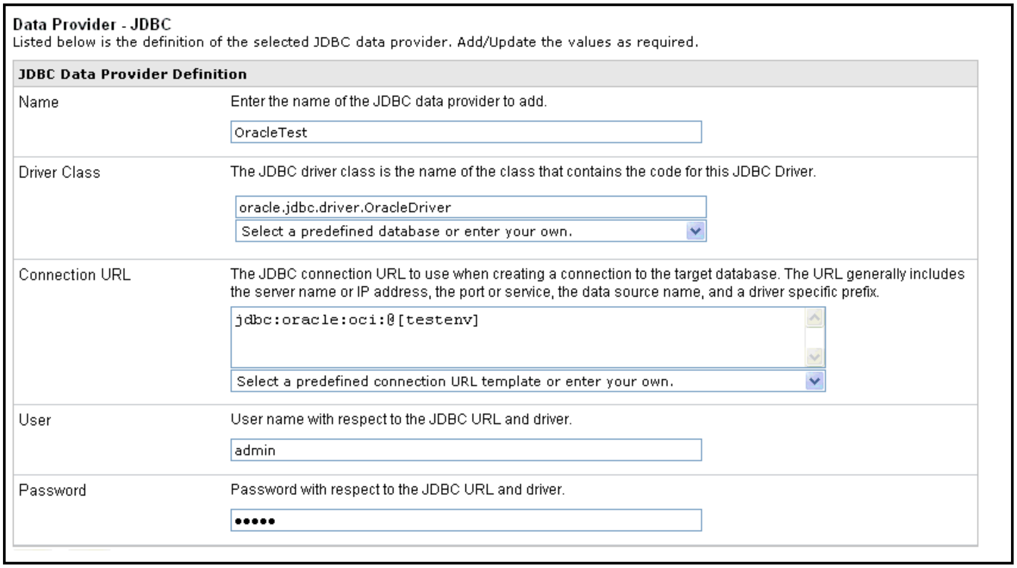
Connection Pooling

Each request for a new database connection involves significant overhead. This can impact performance if obtaining new connections occurs frequently. When connection pooling is enabled, connections are made from connection pools limit the cost of creating connections.

☐ On

1. Click *Add* in the Data Provider Name section.

The Data Provider - JDBC pane opens.



Data Provider - JDBC

Listed below is the definition of the selected JDBC data provider. Add/Update the values as required.

JDBC Data Provider Definition

Name	Enter the name of the JDBC data provider to add. OracleTest
Driver Class	The JDBC driver class is the name of the class that contains the code for this JDBC Driver. oracle.jdbc.driver.OracleDriver Select a predefined database or enter your own.
Connection URL	The JDBC connection URL to use when creating a connection to the target database. The URL generally includes the server name or IP address, the port or service, the data source name, and a driver specific prefix. jdbc:oracle:oci:@[testenv] Select a predefined connection URL template or enter your own.
User	User name with respect to the JDBC URL and driver. admin
Password	Password with respect to the JDBC URL and driver. ●●●●●

2. Enter a name for the JDBC data provider.
3. From the Driver Class drop-down list, select the JDBC class for the data provider.

   You can also manually enter the name.
4. From the Connection URL drop-down list, select the connection URL to use when creating a connection to the target database.

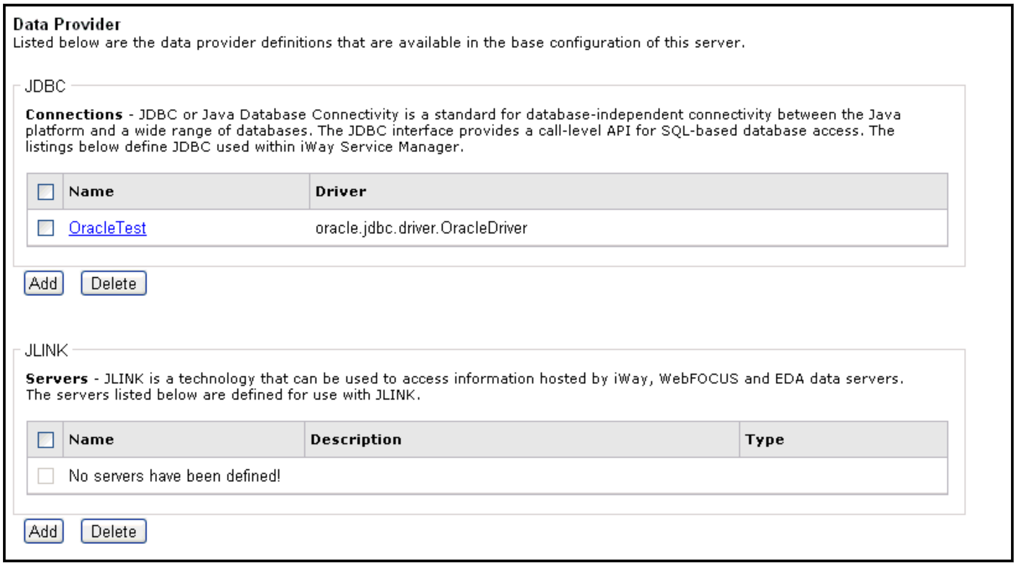
   You can also manually enter the URL.
5. Enter the user ID to access the repository database.
6. Enter the password to access the repository database.
7. Click *Test*.

   You should receive a response that says:

   ```
   The JDBC data provider test completed successfully.
   ```

   If you receive an error, troubleshoot accordingly. Ensure the driver is in the iWay7\lib directory. For more information, see the *iWay Installation and Configuration Guide*.
8. Click *Update* if the test is successful.

You connection appears on the Data Provider pane. If you need to change its parameters, you can click the name of the connection.



Data Provider
Listed below are the data provider definitions that are available in the base configuration of this server.

JDBC

Connections - JDBC or Java Database Connectivity is a standard for database-independent connectivity between the Java platform and a wide range of databases. The JDBC interface provides a call-level API for SQL-based database access. The listings below define JDBC used within iWay Service Manager.

□	Name	Driver
□	OracleTest	oracle.jdbc.driver.OracleDriver

JLINK

Servers - JLINK is a technology that can be used to access information hosted by iWay, WebFOCUS and EDA data servers. The servers listed below are defined for use with JLINK.

□	Name	Description	Type
□	No servers have been defined!		

If you need to define both the target and source repositories, repeat this procedure to define another repository.

Migrating Repositories

You can migrate repositories using the iWay Service Manager Administration Console. These repositories can be for iWay Service Manager, the older iWay Adapter Manager, Servlet iBSP, or iWay Connector for JCA. The structure of the repository has not changed.

Some of the things you can migrate include:

- Migrate the data in the default iWay SM HSQL database to another database repository.
- Migrate an older iWay Adapter Manager repository into the default iWay SM HSQL database.
- Migrate a Servlet iBSP or iWay Connector for JCA database repository.

Note: Monitoring tables are not migrated.

In this section:

- *Source* repository refers to the older existing repository you wish to migrate.
- *Target* repository refers to the new repository you wish to use.

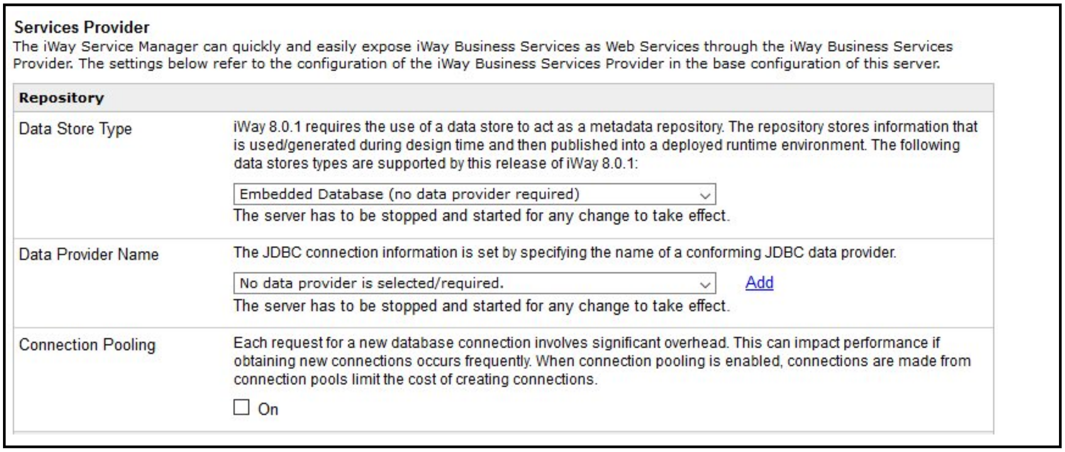
Migration Steps

The following steps are required to migrate a repository:

1. Ensure you have created the new repository tables. For more information, see the *iWay Installation and Configuration Guide*.
2. Ensure the JDBC driver for your target and source repositories are in the iWay7\lib directory. For more information, see the *iWay Installation and Configuration Guide*.
3. Define the source and target repositories as Data Providers using the iWay Service Manager Administration Console as explained in *Add a Data Provider*.
4. Start the migration as explained in *Migrate a Repository*.

Procedure: How to Migrate a Repository

To migrate a repository:



Services Provider

The iWay Service Manager can quickly and easily expose iWay Business Services as Web Services through the iWay Business Services Provider. The settings below refer to the configuration of the iWay Business Services Provider in the base configuration of this server.

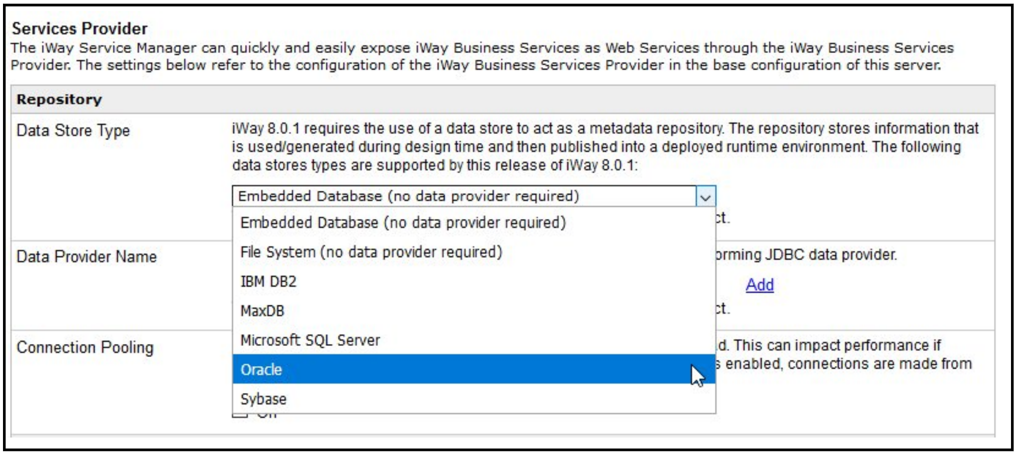
Repository	
Data Store Type	iWay 8.0.1 requires the use of a data store to act as a metadata repository. The repository stores information that is used/generated during design time and then published into a deployed runtime environment. The following data stores types are supported by this release of iWay 8.0.1: Embedded Database (no data provider required) The server has to be stopped and started for any change to take effect.
Data Provider Name	The JDBC connection information is set by specifying the name of a conforming JDBC data provider. No data provider is selected/required. Add The server has to be stopped and started for any change to take effect.
Connection Pooling	Each request for a new database connection involves significant overhead. This can impact performance if obtaining new connections occurs frequently. When connection pooling is enabled, connections are made from connection pools limit the cost of creating connections. ☐ On

1. Click *Services Provider* in the left pane.

   The currently selected Data Store Type and Data Provider Name determine the source repository.

2. Set the source repository by selecting the *Data Store Type* and *Data Provider Name*, and clicking *Update*.

The Data Provider Name is the name you used when you defined the source repository.



Services Provider

The iWay Service Manager can quickly and easily expose iWay Business Services as Web Services through the iWay Business Services Provider. The settings below refer to the configuration of the iWay Business Services Provider in the base configuration of this server.

Repository

Data Store Type: iWay 8.0.1 requires the use of a data store to act as a metadata repository. The repository stores information that is used/generated during design time and then published into a deployed runtime environment. The following data stores types are supported by this release of iWay 8.0.1:

Embedded Database (no data provider required)

Embedded Database (no data provider required)
File System (no data provider required)
IBM DB2
MaxDB
Microsoft SQL Server
Oracle
Sybase

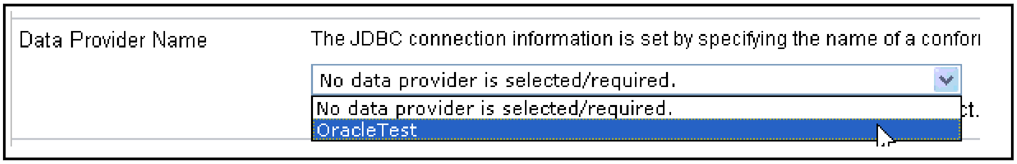
ct.

Data Provider Name: orming JDBC data provider. Add

ct.

Connection Pooling: d. This can impact performance if s enabled, connections are made from

3. Set the target repository by selecting a repository you want to migrate, for example, Oracle, from the Data Store Type drop-down list.



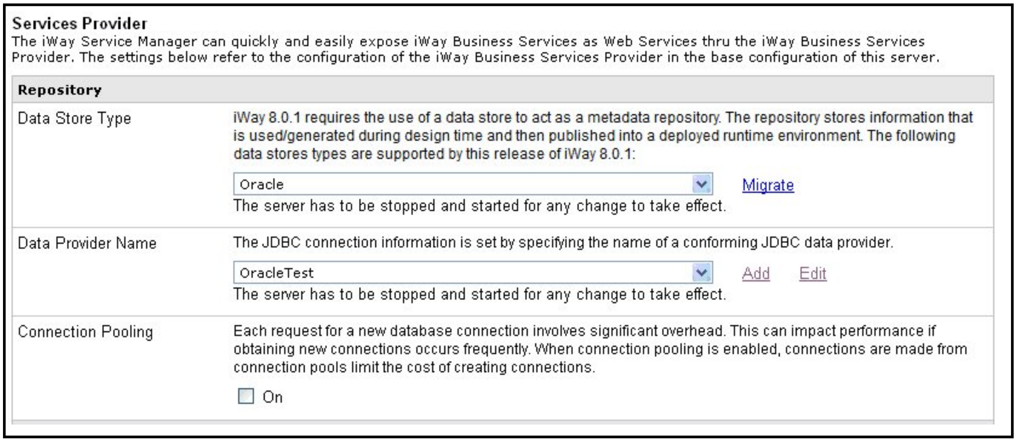
Data Provider Name: The JDBC connection information is set by specifying the name of a confor

No data provider is selected/required.

t.

4. From the Data Provider Name drop-down list, select the name of the data provider (for example, OracleTest).

The Repository section in the Services Provider pane refreshes, as shown in the following image.



Services Provider

The iWay Service Manager can quickly and easily expose iWay Business Services as Web Services thru the iWay Business Services Provider. The settings below refer to the configuration of the iWay Business Services Provider in the base configuration of this server.

Repository

Data Store Type: iWay 8.0.1 requires the use of a data store to act as a metadata repository. The repository stores information that is used/generated during design time and then published into a deployed runtime environment. The following data stores types are supported by this release of iWay 8.0.1:

Oracle Migrate

The server has to be stopped and started for any change to take effect.

Data Provider Name: The JDBC connection information is set by specifying the name of a conforming JDBC data provider.

OracleTest Add Edit

The server has to be stopped and started for any change to take effect.

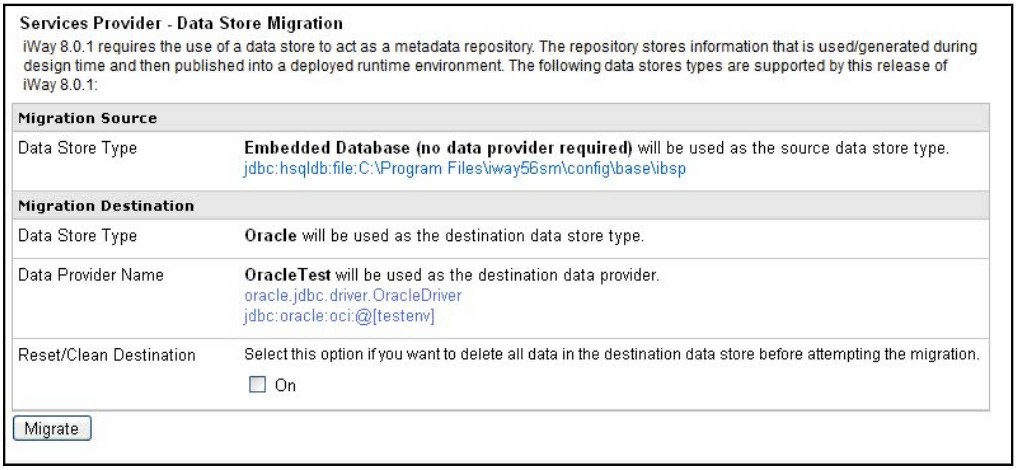
Connection Pooling: Each request for a new database connection involves significant overhead. This can impact performance if obtaining new connections occurs frequently. When connection pooling is enabled, connections are made from connection pools limit the cost of creating connections.

☐ On

Notice the *Migrate* link next to the Data Store Type drop-down list.

5. Click *Migrate*.

   The Services Provider - Data Store Migration pane opens.



Services Provider - Data Store Migration

iWay 8.0.1 requires the use of a data store to act as a metadata repository. The repository stores information that is used/generated during design time and then published into a deployed runtime environment. The following data stores types are supported by this release of iWay 8.0.1:

Migration Source	
Data Store Type	**Embedded Database (no data provider required)** will be used as the source data store type. jdbc:hsqldb:file:C:\Program Files\iway56sm\config\base\ibsp
Migration Destination	
Data Store Type	**Oracle** will be used as the destination data store type.
Data Provider Name	**OracleTest** will be used as the destination data provider. oracle.jdbc.driver.OracleDriver jdbc:oracle:oci:@[testenv]
Reset/Clean Destination	Select this option if you want to delete all data in the destination data store before attempting the migration. □ On

   The table that is provided is divided into two sections:

   - Migration Source - Displays the current repository that is being used.
   - Migration Destination - Displays the destination repository to which you are migrating.

6. Review and verify all the information to make sure it is correct.

   Note: To perform a clean migration, you can select the *On* check box in the Reset/Clean Destination area to delete all data that is currently in the destination repository before proceeding.

7. Click *Migrate*.

   Information about the migration process appears. Ensure there are no critical errors.

   After the migration completes, iWay Business Services Provider is still set to use the source repository. You must set it to use the destination repository instead.

8. Click *Services Provider* on the left.

9. Select the type of repository you wish to use from the Data Store Type drop-down list.

10. Select the connection you just defined from the Data Provider Name drop-down list.

11. Restart iWay Service Manager for your changes to take effect.

Directory Provider

Directory providers offer access to hierarchical maps of information. A directory might be a point in a file system or a registry of information, such as Microsoft's Active Directory. The directory providers offer access to information in a directory for some purpose. For example, an LDAP provider offers generalized access to a directory that responds to the LDAP protocol, while other providers offer directory access for more specific purposes.

For clarity, purpose-specific directory providers appear on the iWay Service Manager Administration Console under their purpose.

In Java and iWay Service Manager, LDAP and Microsoft Active Directory are defined and handled in exactly the same method, through the use of the Directory Provider. Any differences are handled automatically within Provider definitions.

LDAP Directory Providers

A directory is a set of information with similar attributes organized in a logical and hierarchical manner. The protocol accesses LDAP directories, regardless of the form of the directory itself. LDAP sees the directory in a standard manner.

- ❏ A directory is a tree of directory entries.
- ❏ An entry consists of a set of attributes.
- ❏ An attribute has a name (an attribute type or attribute description) and one or more values. The attributes are defined in a schema.
- ❏ Each entry has a unique identifier, its Distinguished Name (DN). This consists of its Relative Distinguished Name (RDN) constructed from some attribute(s) in the entry, followed by the DN of the parent entry. Think of the DN as a full filename and the RDN as a relative filename in a folder.

A DN may change over the lifetime of the entry, for instance, when entries are moved within a tree. To reliably and unambiguously identify entries, a unique key (called a UUID) may be provided in the set of the entry's operational attributes.

iWay provides a simple access function to supply the value of a single LDAP attribute for use as a configuration parameter. The following function accesses the information in the directory:

```
_LDAP( filter, attribute, context [,providername])
```

For example: _LDAP("cn=John Doe",mail,'dc=example') might return the mailing address for John.

LDAP providers may also be used to hold certificates for security operations.

LDAP Directory Provider as a Certificate Store Provider

An LDAP system can also be used to hold certificates for security operations. A configured directory provider pointing to an LDAP system can be used in the configuration of the components that support a certificate store, such as AS2. A certificate store, also called a certstore, is a database of public key certificates and certificate revocation lists. An LDAP server used as a CertStore must support anonymous access.

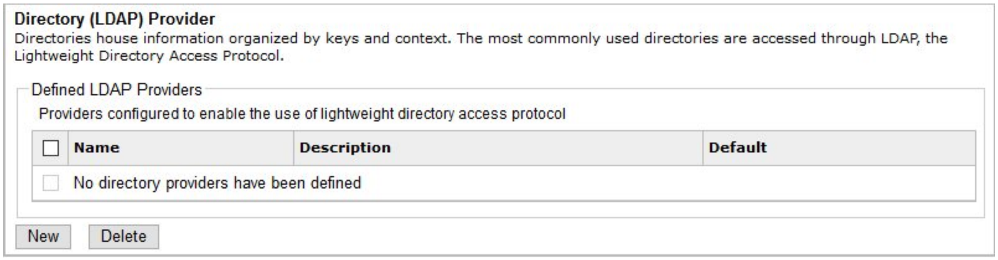
The structure within the LDAP must follow the RFC2587 specifications to store the certificate information. The directory provider configuration screen provides a test facility. This simply verifies that the specified LDAP URL can be used to access the directory. It does not validate security or other attributes of the directory.

Procedure: How to Define a Directory Provider

To define a directory provider:

1. In the left console pane of the Server menu, select *Directory Provider*.

   The Directory Provider pane opens.



Directory (LDAP) Provider

Directories house information organized by keys and context. The most commonly used directories are accessed through LDAP, the Lightweight Directory Access Protocol.

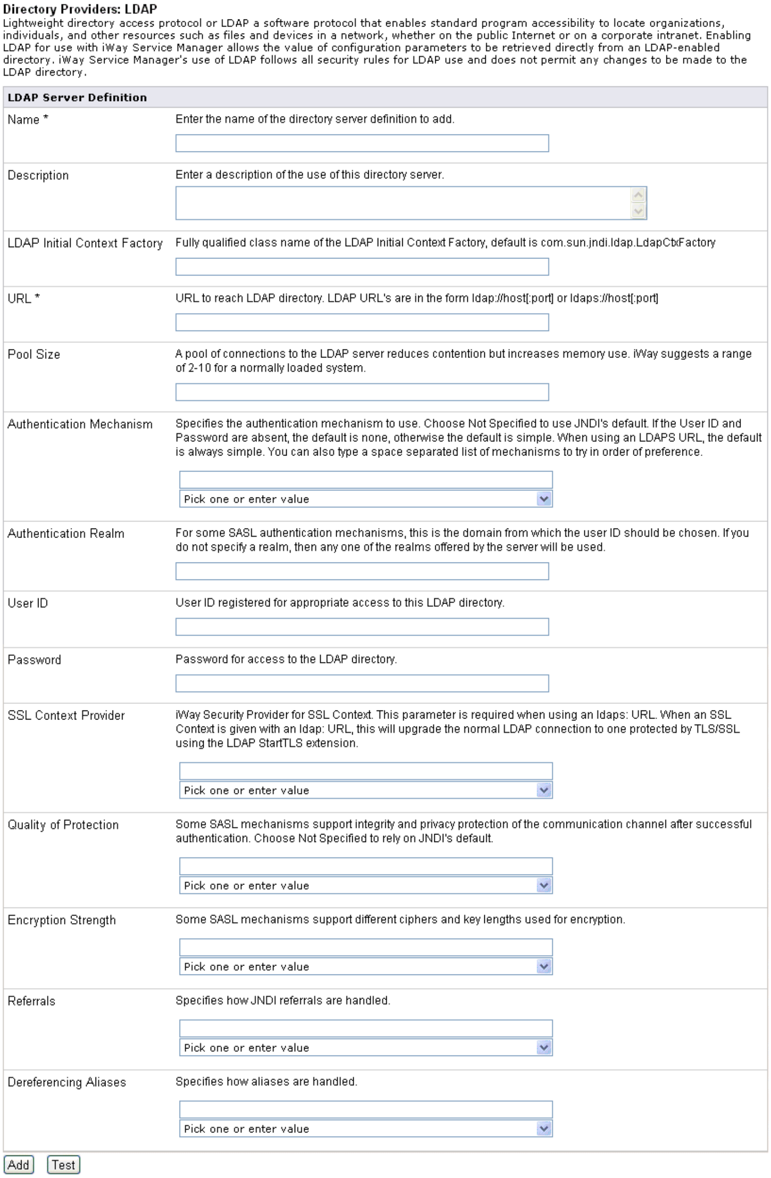
Defined LDAP Providers

Providers configured to enable the use of lightweight directory access protocol

□	Name	Description	Default
□	No directory providers have been defined		

2. Click *New* in the Defined LDAP Providers section.

The Directory Providers: LDAP pane opens.



Directory Providers: LDAP

Lightweight directory access protocol or LDAP a software protocol that enables standard program accessibility to locate organizations, individuals, and other resources such as files and devices in a network, whether on the public Internet or on a corporate intranet. Enabling LDAP for use with iWay Service Manager allows the value of configuration parameters to be retrieved directly from an LDAP-enabled directory. iWay Service Manager's use of LDAP follows all security rules for LDAP use and does not permit any changes to be made to the LDAP directory.

LDAP Server Definition	
Name *	Enter the name of the directory server definition to add.
Description	Enter a description of the use of this directory server.
LDAP Initial Context Factory	Fully qualified class name of the LDAP Initial Context Factory, default is com.sun.jndi.ldap.LdapCtxFactory
URL *	URL to reach LDAP directory. LDAP URL's are in the form ldap://host[:port] or ldaps://host[:port]
Pool Size	A pool of connections to the LDAP server reduces contention but increases memory use. iWay suggests a range of 2-10 for a normally loaded system.
Authentication Mechanism	Specifies the authentication mechanism to use. Choose Not Specified to use JNDI's default. If the User ID and Password are absent, the default is none, otherwise the default is simple. When using an LDAPS URL, the default is always simple. You can also type a space separated list of mechanisms to try in order of preference. Pick one or enter value
Authentication Realm	For some SASL authentication mechanisms, this is the domain from which the user ID should be chosen. If you do not specify a realm, then any one of the realms offered by the server will be used.
User ID	User ID registered for appropriate access to this LDAP directory.
Password	Password for access to the LDAP directory.
SSL Context Provider	iWay Security Provider for SSL Context. This parameter is required when using an ldaps: URL. When an SSL Context is given with an ldap: URL, this will upgrade the normal LDAP connection to one protected by TLS/SSL using the LDAP StartTLS extension. Pick one or enter value
Quality of Protection	Some SASL mechanisms support integrity and privacy protection of the communication channel after successful authentication. Choose Not Specified to rely on JNDI's default. Pick one or enter value
Encryption Strength	Some SASL mechanisms support different ciphers and key lengths used for encryption. Pick one or enter value
Referrals	Specifies how JNDI referrals are handled. Pick one or enter value
Dereferencing Aliases	Specifies how aliases are handled. Pick one or enter value

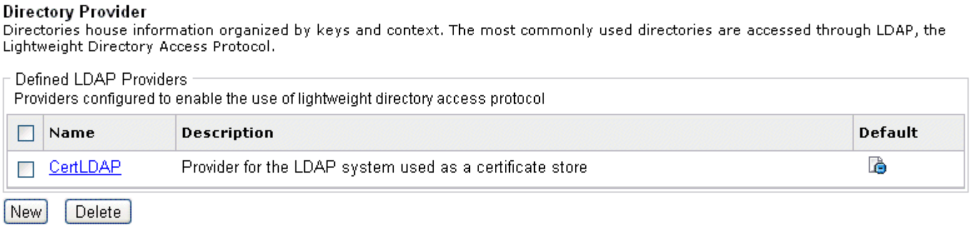
3. Provide the appropriate values for your LDAP connection parameters as listed and defined in the following table.

Parameter	Description
Name *	Enter the name of the directory server definition to add.
Description	Enter a description of the use of this directory server.
LDAP Initial Context Factory	Fully qualified class name of the LDAP Initial Context Factory, default is `com.sun.jndi.ldap.LdapCtxFactory`.
URL *	URL to reach LDAP directory. LDAP URLs are in the form: `ldap://`*`host[:port]`* or `ldaps://`*`host[:port]`*
Pool Size	A pool of connections to the LDAP server reduces contention but increases memory use. iWay suggests a range of 2-10 for a normally loaded system.
Authentication Mechanism	Specifies the authentication mechanism to use. Select *Not Specified* to use the JNDI default. If the User ID and Password are absent, the default is *none*, otherwise the default is *simple*. When using an LDAPS URL, the default is always *simple*. You can also type a space separated list of mechanisms to try in order of preference.
Authentication Realm	For some SASL authentication mechanisms, this is the domain from which the user ID should be chosen. If you do not specify a realm, then any one of the realms offered by the server will be used.
User ID	User ID registered for appropriate access to this LDAP directory.
Password	Password for access to the LDAP directory.
SSL Context Provider	iWay Security Provider for SSL Context. This parameter is required when using an ldaps: URL. When an SSL Context is given with an ldap: URL, this will upgrade the normal LDAP connection to one protected by TLS/SSL using the LDAP StartTLS extension.

Parameter	Description
Quality of Protection	Some SASL mechanisms support integrity and privacy protection of the communication channel after successful authentication. Choose Not Specified to rely on the JNDI default.
Encryption Strength	Some SASL mechanisms support different ciphers and key lengths used for encryption.
Referrals	Specifies how JNDI referrals are handled.
Dereferencing Aliases	Specifies how aliases are handled.
JCE Provider for CertStore	JCE Provider used to create the CertStore when the LDAP provider is used as a CertStore provider

4. Click *Test* to test the connection to the LDAP provider.
5. Click *Add* when you are finished.

   You are returned to the main Directory Provider pane and the new LDAP directory provider that was defined is added to the list.



Directory Provider

Directories house information organized by keys and context. The most commonly used directories are accessed through LDAP, the Lightweight Directory Access Protocol.

Defined LDAP Providers

Providers configured to enable the use of lightweight directory access protocol

□	Name	Description	Default
□	CertLDAP	Provider for the LDAP system used as a certificate store	✓

Note: The LDAP provider is set as the default provider, if it is the first one that is created. When the default provider is removed, the first remaining provider is automatically set as the default.

6. To define multiple directory providers, repeat this procedure.

   A defined directory provider can be used as a named provider during the configuration of supporting components, such as AS2.

Security Provider

A cryptographic system requires a mechanism for the storage and use of cryptographic information. For more information on iSM security and security providers, see the *iWay Service Manager Security Guide*.

A keystore is a collection of keys and certificates. There are two types of keystore entries:

- ❑ **Key Entry.** This type of keystore entry stores sensitive cryptographic key information in a protected format. Typically this is a secret key or a private key with a certificate chain.
- ❑ **Trusted Certificate Entry.** This type of keystore entry contains a single public key certificate belonging to another entity. It is called trusted because the keystore owner trusts that the certificates belongs to the subject (owner) of the certificate.

Entries in a keystore are referred to by their "alias", which is a simple unique string.

A truststore is a keystore used to hold the certificate of trusted Certificate Authorities.

A system may need to use a variety of keystores for different purposes. iWay Service Manager identifies named keystore providers, each of which represents one keystore and the appropriate access credentials and algorithms needed to access the information belonging to that keystore. The keystore provider is identified by name to other components of the system that require access to the security information, such as AS2 or HTTP inlets.

Definition References

A **keystore** is a database of key material. Key material is used for a variety of purposes, including authentication and data integrity. There are various types of keystores available, including "PKCS12" and Sun's "JKS." Some keystores can contain both encryption keys and security certificates. Formally, however, a keystore holds the private key for one or more PKI key pairs.

A **truststore** is a database of key material. It holds the public certificates of trusted partners for message exchange. Although it is possible to share a single file with the keystore, more formally a truststore and a keystore are separate entities.

A **certificate store**, also called a *certstore*, is a database of public key certificates and Certificate Revocation Lists. The CRL is required to stop the use of a certificate when it would otherwise be considered valid. A CRL is not needed to tell you a certificate is bad once its expiration date is reached. In fact, CRLs are usually cleaned of expired CRLs after one complete CRL revision period has elapsed. This means the expired CRL will continue to appear in at most one CRL after it expired.

If certificate revocation is turned on, you will need one CRL for each CA in the certificate chain you want to verify. If a CRL is missing, there is no way to know whether certificates issued by that Certificate Authority are still valid. Therefore, all certificates issued by this CA will be considered revoked. A CRL that contains no certificates is acceptable. It belongs to a Certificate Authority that did not revoke any certificates.

Directory CertStore Providers

CertStore providers define the directories from which certificates and CRLs can be loaded. A configured Directory Certstore provider can be used as a named provider in the components that support CRL checking for messages.

LDAP CertStore Providers

The following section describes LDAP certstore providers.

Adding Debug Information for Certstore

If you encounter issues running the LDAP Certstore provider, you may wish to enable additional debugging. You can add the system property `-Djava.security.debug=certpath` to trace what the Sun CertStores and CertPathBuilder are doing. This will show you more information regarding the way certificates are being loaded and used.

Sun PKIX CertPathBuilder

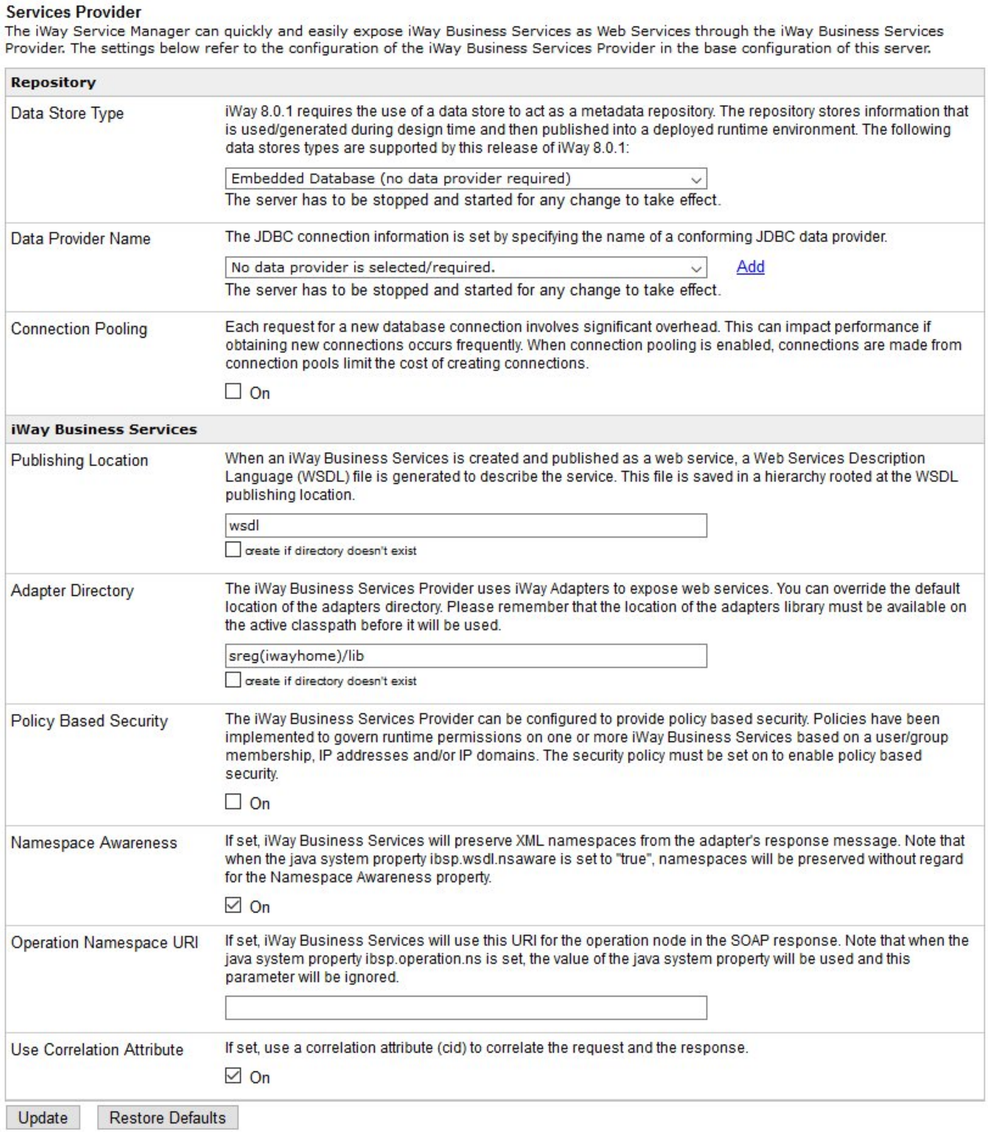
The Sun PKIX CertPathBuilder takes the RFC2587 literally and demands that certificates in the reverse field of the crossCertificatePair be a CA. In particular, it demands that a BasicConstraints extension be present with maxPathLen greater or equal to 0. The End Entity certificates are stored in the crossCertificatePair. The Basic Constraints for an End Entity specifies that this certificate is not a CA and therefore the maxPathLen is -1. This breaks the Sun PKIX CertPathBuilder. As the result, if you send incomplete certificate chains to our AS2 listener, make sure you also set the PKIX JCE Provider to BC to make it work properly.

Procedure: How to Define a CertStore Directory Provider

To define a CertStore directory provider:

1. In the left console pane of the Server menu, select *Security Provider*.

The Security Provider pane opens.



Services Provider

The iWay Service Manager can quickly and easily expose iWay Business Services as Web Services through the iWay Business Services Provider. The settings below refer to the configuration of the iWay Business Services Provider in the base configuration of this server.

Repository

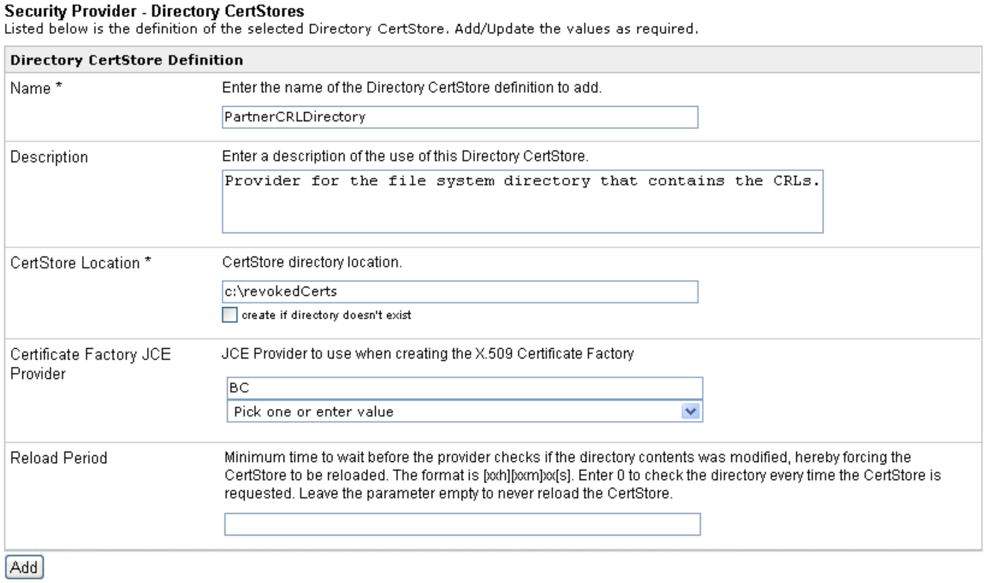
Data Store Type	iWay 8.0.1 requires the use of a data store to act as a metadata repository. The repository stores information that is used/generated during design time and then published into a deployed runtime environment. The following data stores types are supported by this release of iWay 8.0.1: Embedded Database (no data provider required) The server has to be stopped and started for any change to take effect.
Data Provider Name	The JDBC connection information is set by specifying the name of a conforming JDBC data provider. No data provider is selected/required. Add The server has to be stopped and started for any change to take effect.
Connection Pooling	Each request for a new database connection involves significant overhead. This can impact performance if obtaining new connections occurs frequently. When connection pooling is enabled, connections are made from connection pools limit the cost of creating connections. □ On

iWay Business Services

Publishing Location	When an iWay Business Services is created and published as a web service, a Web Services Description Language (WSDL) file is generated to describe the service. This file is saved in a hierarchy rooted at the WSDL publishing location. wsdl □ create if directory doesn't exist
Adapter Directory	The iWay Business Services Provider uses iWay Adapters to expose web services. You can override the default location of the adapters directory. Please remember that the location of the adapters library must be available on the active classpath before it will be used. sreg(iwayhome)/lib □ create if directory doesn't exist
Policy Based Security	The iWay Business Services Provider can be configured to provide policy based security. Policies have been implemented to govern runtime permissions on one or more iWay Business Services based on a user/group membership, IP addresses and/or IP domains. The security policy must be set on to enable policy based security. □ On
Namespace Awareness	If set, iWay Business Services will preserve XML namespaces from the adapter's response message. Note that when the java system property ibsp.wsdl.nsaware is set to "true", namespaces will be preserved without regard for the Namespace Awareness property. ☑ On
Operation Namespace URI	If set, iWay Business Services will use this URI for the operation node in the SOAP response. Note that when the java system property ibsp.operation.ns is set, the value of the java system property will be used and this parameter will be ignored.
Use Correlation Attribute	If set, use a correlation attribute (cid) to correlate the request and the response. ☑ On

2. Click *New* in the Directory CertStore section.

The Directory CertStore Definition pane opens.



Security Provider - Directory CertStores

Listed below is the definition of the selected Directory CertStore. Add/Update the values as required.

Directory CertStore Definition

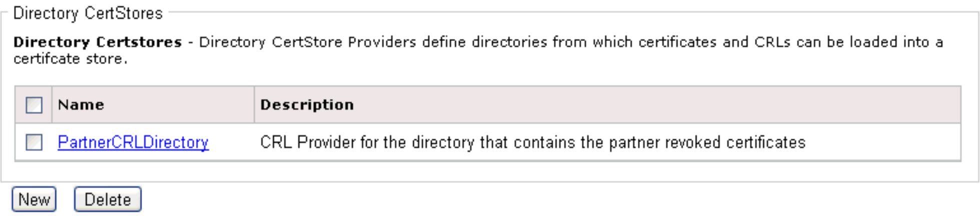
Name *	Enter the name of the Directory CertStore definition to add. PartnerCRLDirectory
Description	Enter a description of the use of this Directory CertStore. Provider for the file system directory that contains the CRLs.
CertStore Location *	CertStore directory location. c:\revokedCerts □ create if directory doesn't exist
Certificate Factory JCE Provider	JCE Provider to use when creating the X.509 Certificate Factory BC Pick one or enter value
Reload Period	Minimum time to wait before the provider checks if the directory contents was modified, hereby forcing the CertStore to be reloaded. The format is [xxh][xxm]xx[s]. Enter 0 to check the directory every time the CertStore is requested. Leave the parameter empty to never reload the CertStore.

3. Provide the appropriate values for your Directory CertStore provider parameters as listed and defined in the following table.

Parameter	Description
Name *	Enter the name of the Directory CertStore definition to add.
Description	Enter a description of the use of this Directory CertStore.
CertStore Location *	CertStore directory location.
Certificate Factory JCE Provider	JCE Provider to use when creating the X.509 Certificate Factory.
Reload Period	Minimum time to wait before the provider checks if the directory contents was modified, hereby forcing the CertStore to be reloaded. The format is [xxh][xxm]xx[s]. Enter 0 to check the directory every time the CertStore is requested. Leave the parameter empty to never reload the CertStore.

4. Click *Add* when you are finished.

You are returned to the main Security Provider pane and the new Directory CertStore provider that was defined is added to the list.



Directory CertStores

Directory Certstores - Directory CertStore Providers define directories from which certificates and CRLs can be loaded into a certifcate store.

□	**Name**	**Description**
□	PartnerCRLDirectory	CRL Provider for the directory that contains the partner revoked certificates

5. To define multiple Directory CertStore providers, repeat this procedure.

   This allows you to configure multiple CertStore directory providers where each can have a different configuration.

   A defined CertStore directory provider can be used as a named provider when configuring components, such as AS2 that support certificate validation.

Keystore Providers

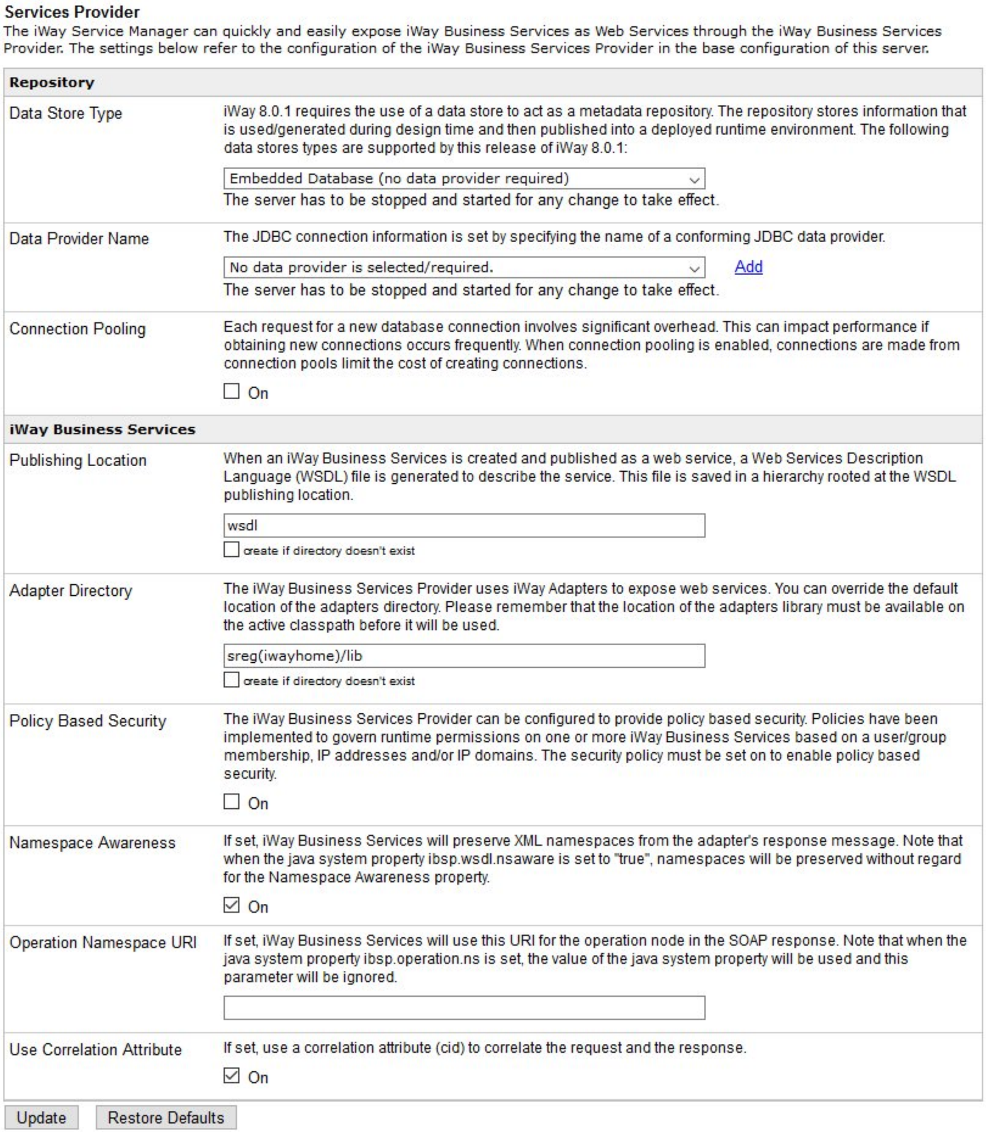
Keystores are standard repositories of security certificates that are used in encryption and operations involving digital signatures. iWay Security Provider configuration supports the creation of multiple keystores that can be used as named providers in the corresponding components, such as AS2 and HTTP. This allows the system to contain multiple types of keystores which may contain different credentials, algorithms and other configurations.

Procedure: How to Define a Keystore Provider

To define a keystore provider using the iWay Service Manager Administration Console:

1. In the left console pane of the Server menu, select *Security Provider*.

The Security Provider pane opens.

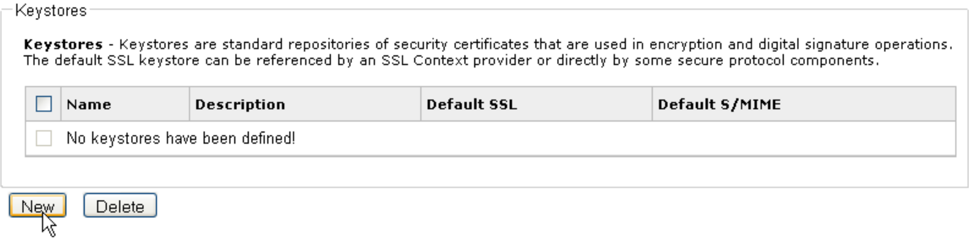


Services Provider

The iWay Service Manager can quickly and easily expose iWay Business Services as Web Services through the iWay Business Services Provider. The settings below refer to the configuration of the iWay Business Services Provider in the base configuration of this server.

Repository	
Data Store Type	iWay 8.0.1 requires the use of a data store to act as a metadata repository. The repository stores information that is used/generated during design time and then published into a deployed runtime environment. The following data stores types are supported by this release of iWay 8.0.1: Embedded Database (no data provider required) The server has to be stopped and started for any change to take effect.
Data Provider Name	The JDBC connection information is set by specifying the name of a conforming JDBC data provider. No data provider is selected/required. Add The server has to be stopped and started for any change to take effect.
Connection Pooling	Each request for a new database connection involves significant overhead. This can impact performance if obtaining new connections occurs frequently. When connection pooling is enabled, connections are made from connection pools limit the cost of creating connections. ☐ On
iWay Business Services	
Publishing Location	When an iWay Business Services is created and published as a web service, a Web Services Description Language (WSDL) file is generated to describe the service. This file is saved in a hierarchy rooted at the WSDL publishing location. wsdl ☐ create if directory doesn't exist
Adapter Directory	The iWay Business Services Provider uses iWay Adapters to expose web services. You can override the default location of the adapters directory. Please remember that the location of the adapters library must be available on the active classpath before it will be used. sreg(iwayhome)/lib ☐ create if directory doesn't exist
Policy Based Security	The iWay Business Services Provider can be configured to provide policy based security. Policies have been implemented to govern runtime permissions on one or more iWay Business Services based on a user/group membership, IP addresses and/or IP domains. The security policy must be set on to enable policy based security. ☐ On
Namespace Awareness	If set, iWay Business Services will preserve XML namespaces from the adapter's response message. Note that when the java system property ibsp.wsdl.nsaware is set to "true", namespaces will be preserved without regard for the Namespace Awareness property. ☑ On
Operation Namespace URI	If set, iWay Business Services will use this URI for the operation node in the SOAP response. Note that when the java system property ibsp.operation.ns is set, the value of the java system property will be used and this parameter will be ignored.
Use Correlation Attribute	If set, use a correlation attribute (cid) to correlate the request and the response. ☑ On

2. Click *New* in the Keystores section.

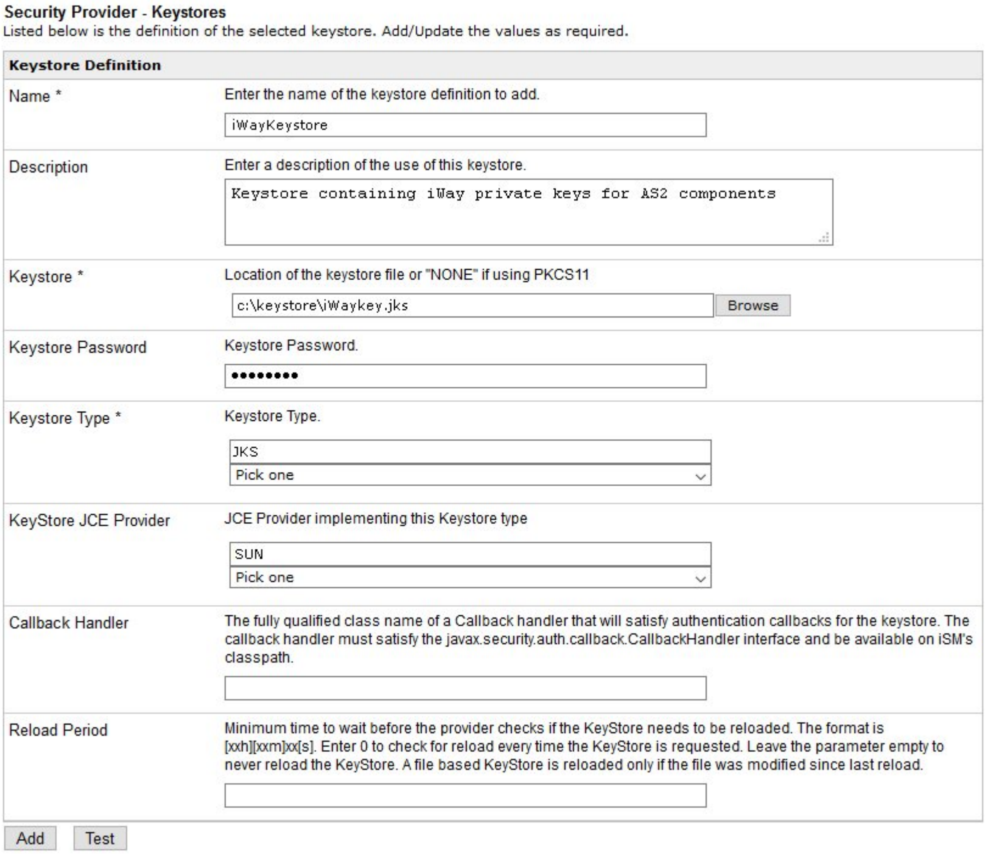


Keystores

Keystores - Keystores are standard repositories of security certificates that are used in encryption and digital signature operations. The default SSL keystore can be referenced by an SSL Context provider or directly by some secure protocol components.

□	Name	Description	Default SSL	Default S/MIME
□	No keystores have been defined!			

The Keystore Definition pane opens.



Security Provider - Keystores

Listed below is the definition of the selected keystore. Add/Update the values as required.

Keystore Definition

Name *	Enter the name of the keystore definition to add. iWayKeystore
Description	Enter a description of the use of this keystore. Keystore containing iWay private keys for AS2 components
Keystore *	Location of the keystore file or "NONE" if using PKCS11 c:\keystore\iWaykey.jks
Keystore Password	Keystore Password. ••••••••
Keystore Type *	Keystore Type. JKS Pick one
KeyStore JCE Provider	JCE Provider implementing this Keystore type SUN Pick one
Callback Handler	The fully qualified class name of a Callback handler that will satisfy authentication callbacks for the keystore. The callback handler must satisfy the javax.security.auth.callback.CallbackHandler interface and be available on iSM's classpath.
Reload Period	Minimum time to wait before the provider checks if the KeyStore needs to be reloaded. The format is [xxh][xxm]xx[s]. Enter 0 to check for reload every time the KeyStore is requested. Leave the parameter empty to never reload the KeyStore. A file based KeyStore is reloaded only if the file was modified since last reload.

3. Enter the parameters for the keystore and make sure to select the appropriate values from the Keystore Type and the Keystore Provider drop-down lists that will correspond to your keystore configuration.

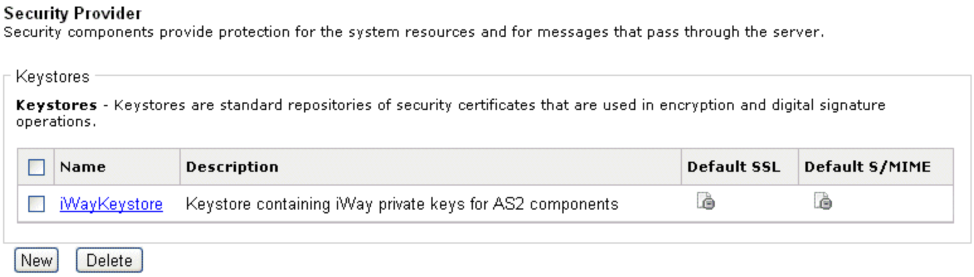
4. In the Callback Handler field, optionally enter the fully qualified name of a callback handler that will satisfy authentication callbacks for the keystore.

   The callback handler must satisfy the *javax.security.auth.callback.CallbackHandler* interface and be available in the classpath for iWay Service Manager.

   It is recommended that developers make use of the Test button to verify that the property values are tested against the Keystore before completing the definition. When there are no failures encountered during testing, then continue from this point.

5. Click *Add*.

   You are returned to the main Security Provider pane and the new keystore that was defined is added to the list.



Security Provider

Security components provide protection for the system resources and for messages that pass through the server.

Keystores

Keystores - Keystores are standard repositories of security certificates that are used in encryption and digital signature operations.

□	Name	Description	Default SSL	Default S/MIME
□	iWayKeystore	Keystore containing iWay private keys for AS2 components	✓	✓

   Note: The keystore provider is set as the default SSL provider and default S/MIME provider if it is the first one that is created. When an SSL or S/MIME default provider is removed, the first remaining keystore provider is automatically set as the default.

6. To define multiple keystore providers, repeat this procedure.

   A defined keystore provider can be used as a named provider when configuring other iWay components (for example, listeners, services, emitters, and so on).

SSL Context Providers

Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols that provide security and data integrity for communications over TCP/IP networks, such as the Internet. These protocols allows client/server applications to communicate across a network in a manner designed to prevent eavesdropping, tampering, and message forgery. TLS and SSL encrypt the segments of network connections at the Transport Layer end-to-end. In cases in which iSM requires SSL or TLS support, the appropriate component requests the name of an SSL Context Provider.

In the typical usage, authentication is unilateral; only the server is authenticated to the end point. That means that the client is aware and sure of the identity of the server but not vice versa. These protocols also support bilateral authentication, in which both the client and the server exchange certificates and are aware of the others identity. This is common in business interactions. Identification is accomplished by the exchange of signed certifications containing the URL, name and address of the end point that sends the certificate. The certificates are in turn signed by a trusted Certificate Authority.

Once defined, an SSL Context Provider can be associated with one or more components (server, such as nHTTP or client, such as an nHTTP emit agent) using SSL or TLS. This is done be naming the provider in the component's configuration. The SSL Context Provider, in turn, relies on keystore and trust store providers that have been previously configured. The provider manages the connections and handshakes between end points, and attempts to optimize connection reuse where possible and consistent with communications security.

If you are configuring an SSL Context Provider to be used for server side, you will need a Keystore Provider as the source of your public certificate which will be recognized by the client. If you configure your SSL Context Provider to require client authentication, you will need a Truststore Provider as the source of the trusted client certificates.

If you are configuring an SSL Context Provider to be used on the client side, you will need a Keystore Provider as the source of your public certificate, and a TrustStore Provider, as the source of the certificates for the servers to be trusted.

An SSL Context Provider requires that both be configured, even though both may not be required. We cannot tell for what the Provider will be utilized. However if your application does not require a value in either Keystore, or Truststore, the contents are not used. The format of Keystore and Truststore must be correct.

TLS protocol version 2 is supported. Basic SSL is no longer considered to be sufficiently secure, and many of its shortcomings have been addressed with TLS. The SSL level setting represents the minimum acceptable security algorithm. iWay Software strongly recommends that TLS be considered as the minimum acceptable level. For secure transactions, specification of TLS version 2 is recommended, provided that both sides of the transaction have this algorithm available. iSM will negotiate for the highest level available when connections are established.

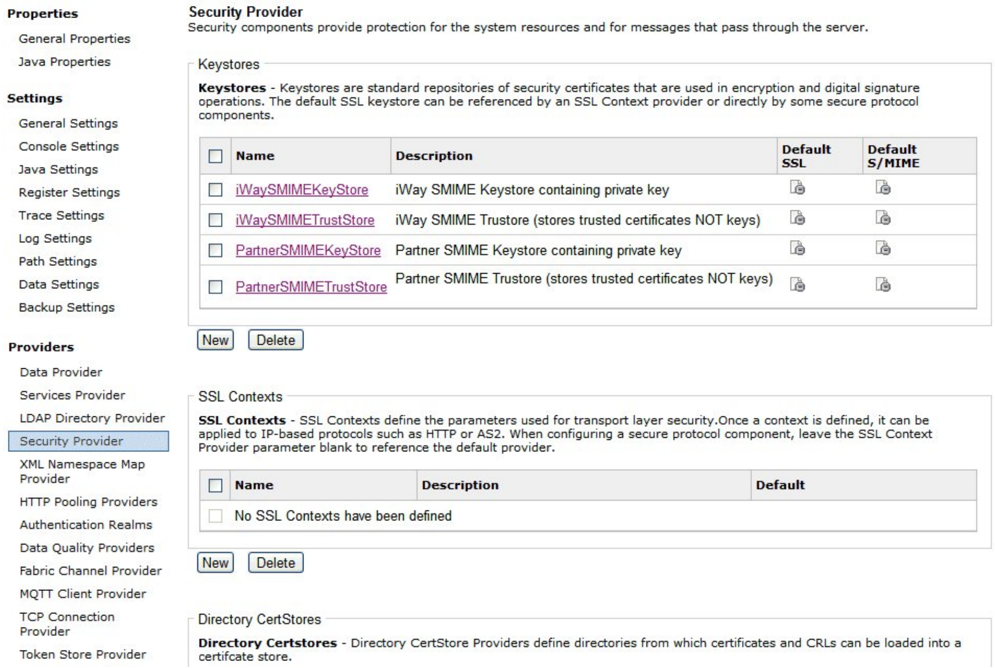
Note: You must have Java version 1.7 configured on your system to use TLS protocol version 2.

Procedure: How to Define an SSL Context Provider

To define an SSL context provider using the iWay Service Manager Administration Console:

1. In the left console pane of the Server menu, select *Security Provider*.

   The Security Provider pane opens.



Security Provider
Security components provide protection for the system resources and for messages that pass through the server.

Keystores

Keystores - Keystores are standard repositories of security certificates that are used in encryption and digital signature operations. The default SSL keystore can be referenced by an SSL Context provider or directly by some secure protocol components.

□	Name	Description	Default SSL	Default S/MIME
□	iWaySMIMEKeyStore	iWay SMIME Keystore containing private key	🗎	🗎
□	iWaySMIMETrustStore	iWay SMIME Trustore (stores trusted certificates NOT keys)	🗎	🗎
□	PartnerSMIMEKeyStore	Partner SMIME Keystore containing private key	🗎	🗎
□	PartnerSMIMETrustStore	Partner SMIME Trustore (stores trusted certificates NOT keys)	🗎	🗎

SSL Contexts

SSL Contexts - SSL Contexts define the parameters used for transport layer security.Once a context is defined, it can be applied to IP-based protocols such as HTTP or AS2. When configuring a secure protocol component, leave the SSL Context Provider parameter blank to reference the default provider.

□	Name	Description	Default
□	No SSL Contexts have been defined		

Directory CertStores

Directory Certstores - Directory CertStore Providers define directories from which certificates and CRLs can be loaded into a certifcate store.

2. Click *New* in the SSL Contexts section.

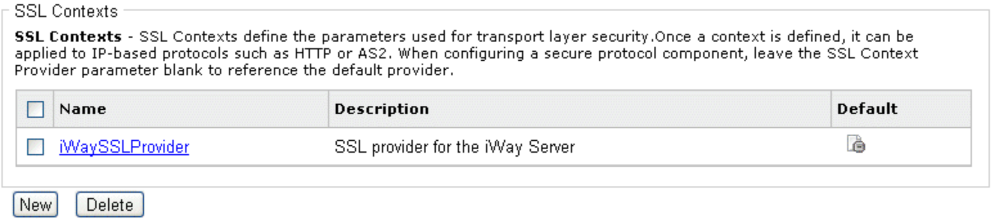
   The SSL Context Provider pane opens.

3. Enter the appropriate values for the SSL context provider parameters.

   For more information, see *Parameters for SSL Context Providers*.

4. Click *Add* when you are finished.

You are returned to the main Security Provider pane and the new SSL context provider that was defined is added to the list.



SSL Contexts

SSL Contexts - SSL Contexts define the parameters used for transport layer security.Once a context is defined, it can be applied to IP-based protocols such as HTTP or AS2. When configuring a secure protocol component, leave the SSL Context Provider parameter blank to reference the default provider.

□	Name	Description	Default
□	iWaySSLProvider	SSL provider for the iWay Server	✓

Note: The SSL context provider is set as the default provider, if it is the first one that is created. When the default provider is removed, the first remaining provider is automatically set as the default.

5. To define multiple SSL context providers, repeat this procedure.

   A defined SSL context provider can be used as a named provider when configuring IP-based components, such as AS2 and HTTP.

   Note: To activate any new security providers that have been configured, you must restart iWay Service Manager.

Reference: Parameters for SSL Context Providers

The following tables list and describes all of the available parameters for SSL context providers.

Property	Description
Name	The name of the SSL Context definition to add.
Description	A brief description of the use of this SSL Context.
Keystore Provider	Configured Security Provider for the keystore you wish to use for this SSL context. Choose *default* to use the default SSL Keystore Provider. Keystores hold private keys.
Truststore Provider	Configured Security Provider for the truststore you wish to use for this SSL context. Choose *default* to use the default SSL Keystore Provider. Truststores hold the certificate of Trusted CAs used to verify peer certificates.

Property	Description
Security Protocol	Specify the version of security protocol that should be used. During SSL handshake, a negotiation selects the protocol to be used from the best mutually supported. This field sets the minimum acceptable security protocol. If the handshake cannot select a mutually supported protocol, the connection fails. The options are: SSL, SSLv2, SSLv3, TLS, TLSv1, and TLSv2. **Note:** You must have Java version 1.7 configured on your system to use TLS protocol version 2.
JCE SSL Context Provider	JCE Provider for the SSL Context.
Enabled Cipher Suites	If supplied, only cipher suites on this list will be enabled for SSL sockets or SSL engines created using this provider. The user must take care that enabled cipher suites are supported by other components specified. Enter as comma-delimited list or use FILE() function. If left blank, all available cipher suites will be enabled and be available during SSL negotiation.

Session Reuse

Property	Description
Session Cache Size	The maximum number of SSL sessions that will be retained in the session cache. Sessions in the cache can be reconnected with less overhead than those not cached.
Session Timeout	Maximum length of time (in seconds) that an SSL session can remain in the cache.

Advanced Server Side

Property	Description
Server Key Alias	Alias for the key to be used to identify secure servers using this SSL context. If not supplied, the key will be selected using JSSE default behavior.
Client Authentication	If true, servers using this provider will use SSL client authentication, that is, the server must receive and authenticate a certificate from the client as part of the SSL handshake.

Advanced Client Side

Property	Description
Client Key Alias	Alias for the key to be used to identify secure clients using this SSL context. If not supplied, the key will be selected using JSSE default behavior.
Hostname Verification	If true, client SSL connections using this provider will attempt to verify that the server certificate matches its host name.

Certificate Revocation Management

Property	Description
Enable Certificate Revocation	Enable CRL checking of certificates during handshake.
OCSP Responder	Name of the OCSP Responder provider. This verifies the status of certificates online instead of relying on Certificate Revocation Lists (CRLs).
JCE PKIX Trust Manager Provider	JCE provider to construct PKIX Trust Manager. Choose 'Not Specified' for default.

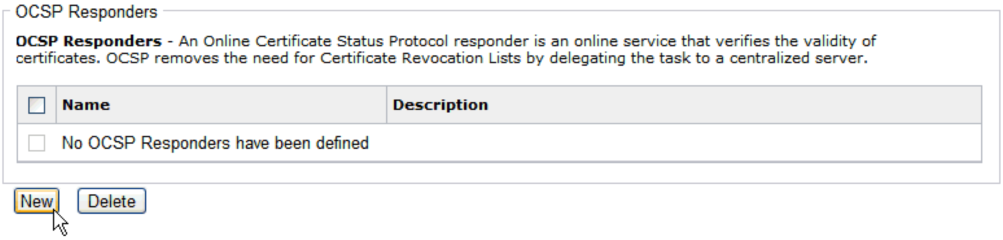
Property	Description
JCE Signature Provider	JCE provider used to verify digital certificate signatures during handshake.
PKIX Certificate Store	Certificate store from which certificate revocation lists are loaded.

OCSP Responder Provider

The Online Certificate Status Protocol (OCSP) is an Internet protocol used to obtain the revocation status of an X.509 digital certificate. It is formalized in RFC 2560.

OCSP was created as an alternative to Certificate Revocation Lists (CRLs), specifically addressing certain problems associated with using CRLs in a Public Key Infrastructure (PKI).The request and response nature of these messages lead to OCSP servers being termed OCSP responders. iSM can communicate with an OCSP responder to obtain the revocation status of a certificate, avoiding the need to manage certificates locally in many cases.

To create an OCSP Responder Provider, navigate to the OCSP Responders section in the Security Provider pane and click New, as shown in the following image.



OCSP Responders

OCSP Responders - An Online Certificate Status Protocol responder is an online service that verifies the validity of certificates. OCSP removes the need for Certificate Revocation Lists by delegating the task to a centralized server.

☐	Name	Description
☐	No OCSP Responders have been defined	

The following table lists the OCSP Responder Provider properties.

Property	Description
Name *	The name of the OCSP Responder definition to add.
Description	A brief description of the use of this OCSP Responder.
Responder URL *	Location of the OCSP responder. For example: `http://ocsp.example.net:80`

Property	Description
Certificate Subject Name	Subject name of the certificate for the OCSP responder. For example, CN=OCSP Responder and O=XYZ Corp.
Certificate Issuer Name	Issuer name of the certificate for the OCSP responder. For example, CN=Enterprise CA and O=XYZ Corp. This property is required if a value for the Certificate Subject Name parameter is not specified.
Certificate Serial Number	Serial number of the OCSP responder's certificate. For example 12345678901234567899. This property is required if a value for the Certificate Subject Name parameter is not specified.
Certificate Store *	Certificate store where the responder certificate can be retrieved.
HTTP Client Provider *	HTTP client provider that manages outgoing connections to the responder.

XML Digital Signature JCE Providers

The XML digital signature agents use the services of the XML digital signature JCE provider. This provider is a standard part of JDK 1.7 and requires no special installation instructions when running with JDK 1.7.

If you are using JDK 1.5, you must add the javax.xml.crypto.jar file to your class path. iWay Software produced this file by compiling (with JDK 1.5) the subset of JDK 1.6 sources that deal with XML digital signatures. You must also declare the provider in jre/lib/security/java.security using a line with the following form:

```
security.provider.N=org.jcp.xml.dsig.internal.dom.XMLDSigRI
```

where:

N

Is the highest provider number already present plus one.

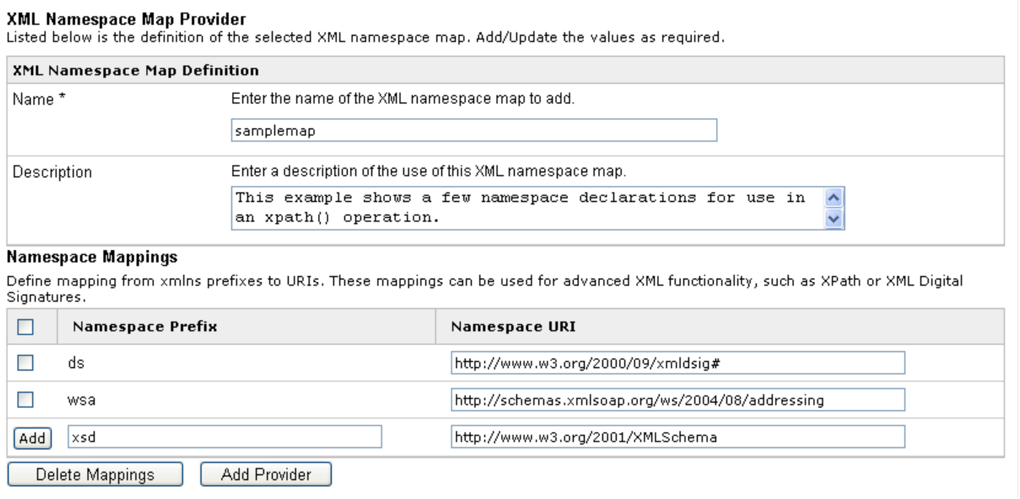
XML Namespace Map Providers

The XML namespace map provider is used to map XML namespace prefixes to the XML namespace URIs. It is also possible to declare multiple XML namespace map providers. Each provider can have any number of namespace declarations. The agents that need to know namespace prefixes have a parameter where you can enter a provider name to declare the namespaces. This can be used to allow namespace prefixes in XPATH expressions like /soapenv:Envelope/soapenv:Header/wsse:Security, or it can be used in reverse to choose which prefix to use when generating new XML content.

The following table lists a sample set of XML namespaces that can be declared with typical prefixes:

Namespace Prefix	Namespace URI
`ds`	`http://www.w3.org/2000/09/xmldsig#`
`ec`	`http://www.w3.org/2001/10/xml-exc-c14n#`
`saml`	`urn:oasis:names:tc:SAML:1.0:assertion`
`soapenv`	`http://schemas.xmlsoap.org/soap/envelope/`
`wsa`	`http://schemas.xmlsoap.org/ws/2004/08/addressing`
`wsse`	`http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd`
`wsu`	`http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd`
`xsd`	`http://www.w3.org/2001/XMLSchema`
`xsi`	`http://www.w3.org/2001/XMLSchema-instance`

You can access the XML Namespace Map Provider window in the iWay Service Manager Administration Console to construct an XML namespace map.



XML Namespace Map Provider

Listed below is the definition of the selected XML namespace map. Add/Update the values as required.

XML Namespace Map Definition	
Name *	Enter the name of the XML namespace map to add. samplemap
Description	Enter a description of the use of this XML namespace map. This example shows a few namespace declarations for use in an xpath() operation.

Namespace Mappings

Define mapping from xmlns prefixes to URIs. These mappings can be used for advanced XML functionality, such as XPath or XML Digital Signatures.

□	Namespace Prefix	Namespace URI
□	ds	http://www.w3.org/2000/09/xmldsig#
□	wsa	http://schemas.xmlsoap.org/ws/2004/08/addressing
Add	xsd	http://www.w3.org/2001/XMLSchema

In this example, two XML namespace maps have been added and a third one is in the process of being defined. When you are finished, click *Add Provider* to make the XML namespace map available.

Once the map is complete, the following XPATH function will locate the y element in a document where the wsa namespace defined in the map matches the namespace URI for that element in the actual document:

```
xpath('/x/wsa:y', samplemap)
```

Note: It is not required for the namespace prefix in the map to match the actual namespace prefix used in the target document.

Pooling Providers

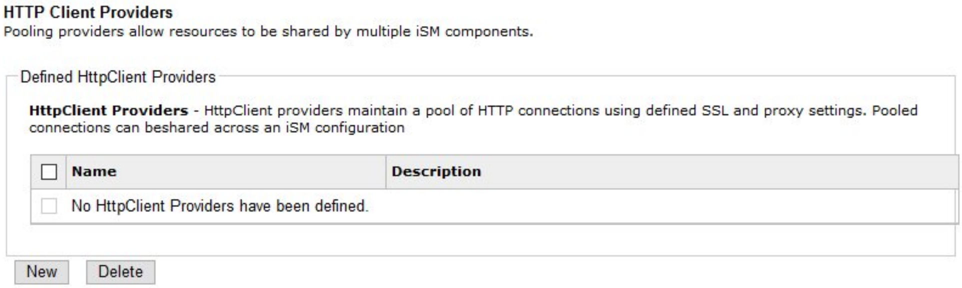
Pooling providers allow HTTP connections to be shared among iWay Service Manager (iSM) components. An instance of the Pooling provider represents a pool of connections. Connections in this pool can share proxy settings, local interface bindings, and, for HTTPS, a single SSL socket factory. Currently, the provider is used by the nHTTP and nAS2 emitters, as well as by the nAS2 MDN subsystem for sending asynchronous MDN messages through HTTP.

Procedure: How to Define Pooling Providers

To define a Pooling provider:

1. In the left console pane of the Server menu, select *Pooling Providers*.

The Pooling Providers pane opens, as shown in the following image.



HTTP Client Providers

Pooling providers allow resources to be shared by multiple iSM components.

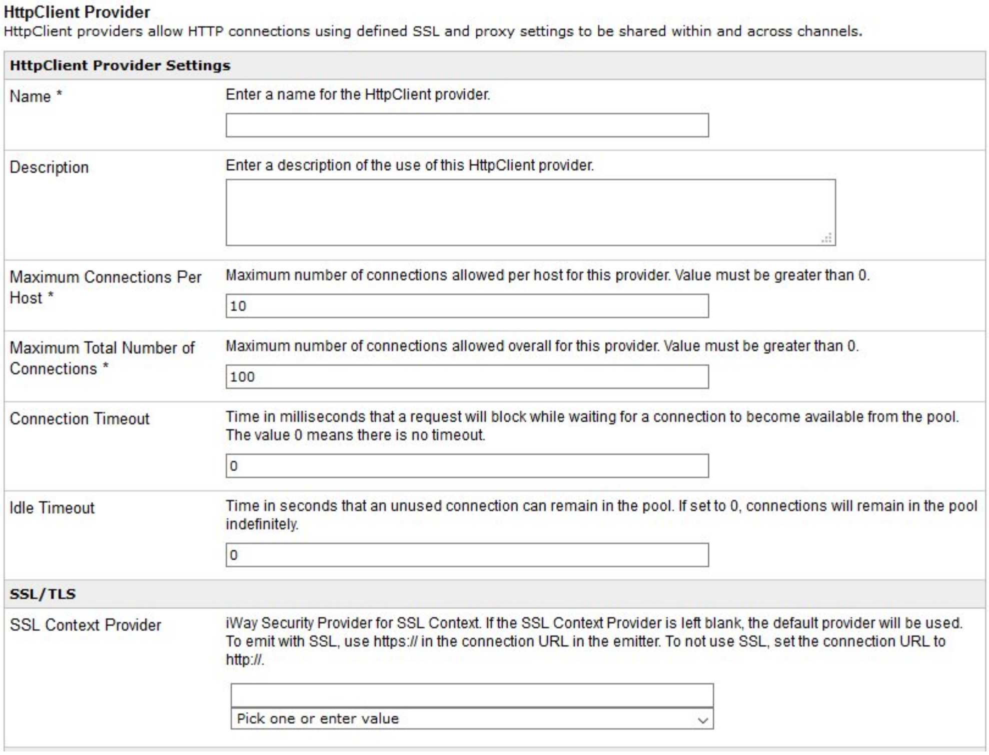
Defined HttpClient Providers

HttpClient Providers - HttpClient providers maintain a pool of HTTP connections using defined SSL and proxy settings. Pooled connections can beshared across an iSM configuration

□	Name	Description
□	No HttpClient Providers have been defined.	

2. Click *New* in the Defined HttpClient Providers section.

   The HttpClient Provider pane opens, which contains settings for the Pooling provider, as shown in the following image.



HttpClient Provider

HttpClient providers allow HTTP connections using defined SSL and proxy settings to be shared within and across channels.

HttpClient Provider Settings	
Name *	Enter a name for the HttpClient provider.
Description	Enter a description of the use of this HttpClient provider.
Maximum Connections Per Host *	Maximum number of connections allowed per host for this provider. Value must be greater than 0. 10
Maximum Total Number of Connections *	Maximum number of connections allowed overall for this provider. Value must be greater than 0. 100
Connection Timeout	Time in milliseconds that a request will block while waiting for a connection to become available from the pool. The value 0 means there is no timeout. 0
Idle Timeout	Time in seconds that an unused connection can remain in the pool. If set to 0, connections will remain in the pool indefinitely. 0
SSL/TLS	
SSL Context Provider	iWay Security Provider for SSL Context. If the SSL Context Provider is left blank, the default provider will be used. To emit with SSL, use https:// in the connection URL in the emitter. To not use SSL, set the connection URL to http://. Pick one or enter value

3. Enter the appropriate values for the Pooling provider parameters.
4. Click *Add* when you are finished.

You are returned to the main Pooling Providers pane and the new provider that was defined is added to the list.

5. To define multiple Pooling providers, repeat this procedure.

Reference: Pooling Provider Parameters

The following tables list and describes all of the available parameters for the Pooling provider.

Parameter	Description
Name	Name for the Pooling provider.
Description	Brief description of the use of this Pooling provider.
Maximum Connections Per Host	Defines the maximum number of simultaneous connections allowed per host. When this threshold is reached, new connections will not be accepted until current connections are closed and the total number of connections is below the limit. Leave this field blank (default) or set a value of zero to have no maximum limit of connections.
Maximum Total Number of Connections	Defines the maximum number of simultaneous connections that are allowed overall. When this threshold is reached, new connections will not be accepted until current connections are closed and the total number of connections is below the limit. Leave this field blank (default) or set a value of zero to have no maximum limit of connections.
Connection Timeout	Maximum length of time (in milliseconds) that a request will block while waiting for a connection to become available from the pool. A value of zero (0) means there is no timeout.
Idle Timeout	Time in seconds that an unused connection can remain in the pool. If set to 0, connections will remain in the pool indefinitely.

SSL/TLS

Parameter	Description
SSL Context Provider	Named iWay Security provider for SSL Context. Defaults to the value assigned to the SSL Context Provider.

Authentication

Parameter	Description
Authentication Preference	If several schemes are returned in the WWW-Authenticate header, this parameter defines which schemes take precedence over others. The value is a comma-separated list of authentication scheme names with the most preferred scheme listed first. The default is negotiate,NTLM,Digest,Basic where negotiate means SPNEGO. Kerberos requires the negotiate scheme and HttpClient version 4.1 or higher.
Kerberos Login Entry	The Application Login Entry in the JAAS login configuration file that will be used to login to Kerberos. This login entry should configure a Kerberos login module (Krb5LoginModule).
Preemptive Basic Authentication	Sends the basic authentication credentials pre-emptively before the server returns an authorization request. This saves a round trip at the risk of significant security issues, such as possibly sending credentials in clear to an unauthorized third party. This property controls the default behavior when the value is not configured explicitly in the emitter.

Cookies

Parameter	Description
Cookie Specification	The cookie management specification determines the rules for parsing, validating, and formatting cookies. By default, *best-match* is selected from the drop-down list, which is the recommended policy. HttpClient version 4.1 or higher is required.

Parameter	Description
Persistent Cookie Store	If set to *true*, cookies are preserved between server reboots. By default, *false* is selected.

Proxy

Parameter	Description
Proxy	If set to *true*, emit through a proxy server.
Proxy User ID	User ID for the proxy challenges.
Proxy Password	Password to access the proxy server.
Proxy Domain	Domain for NTLM proxy authentication.
Proxy Host	Host where the proxy can be accessed.
Proxy Port	Port where the proxy can be accessed.

Advanced

Parameter	Description
Set TCP No Delay	If set to *true*, Nagle's Algorithm on the client socket is disabled. This will result in faster line turnaround at the expense of an increased number of packets.

Parameter	Description
Reuse Socket Address	Allow sockets to be bound to the local addresses of sockets that already closed, but are still in a TIME_WAIT state. In applications that make large numbers of short-lived connections, this can reduce or eliminate *address in use* and related errors when no more ephemeral ports are available to be bound. When using this option, you should specify the *any local address* wildcard (0.0.0.0 in case of IPv4 or :: for IPv6) as the value of the IP Interface Host parameter below. For more detailed information and a discussion on the interactions between the Reuse Socket Address and IP Interface Host parameters, see the following website *http://stackoverflow.com/questions/14388706/socket-options-so-reuseaddr-and-so-reuseport-how-do-they-differ-do-they-mean-t* **Note:** The effects of setting this option can be complex and should be handled with care.
Linger-On-Close Timeout	This option disables or enables immediate return from a close() function of a TCP Socket. Enabling this option with a non-zero integer timeout means that a close() function will block pending the transmission and acknowledgment of all data written to the peer, at which point the socket is closed gracefully. Upon reaching the linger timeout, the socket is closed forcefully, with a TCP RST. Enabling the option with a timeout of zero does a forceful close immediately. Note that this may have the effect of leaving sockets on the server side in a wait state. Enter -1 or leave blank for the proper default. **Note:** The effects of setting this option can be complex and should be handled with care.
Redirect Strategy	Determines how an HTTP redirect response is handled. The default is to follow redirects according to the RFC. The off strategy does not redirect any request.
IP Interface Host	Local IP Interface from which the outgoing IP socket originates.

Cookie Management

By default, the Pooling provider handles cookies automatically. The provider looks for cookies in the responses it receives and then stores them in a cookie store. When a new request is made, the provider looks in the cookie store and the cookies that match are resubmitted to the originating server. Each Pooling provider manages its own independent cookie stores.

Cookie management is configured with parameters on the Pooling provider. The Cookie Specification parameter determines the rules for parsing, validating, and formatting cookies. The specifications that are available for this parameter are listed and described in the following table.

Cookie Specification	Description
standard	Relaxed profile RFC6265 section
standard-strice	Well-behaved profle Relaxed profile RFC6265 section
best-match	Selects a cookie policy based on the format of cookies sent with the HTTP response.
rfc2109	Cookie Version 0.
rfc2965	Cookie Version 0 and Cookie2 Version 1.
compatibility	Closely mimics (mis)behavior of common web browsers.
netscape	Conforms to the original draft specification published by Netscape Communications. Should be avoided unless strictly necessary for legacy code.
ignoreCookies	Cookie management is disabled.

The recommended Cookie Specification is *best-match* because it can handle cookie versions 0, 1, and 2.

To disable automatic cookie management, select *ignoreCookies*. Cookies will continue to appear in the request header namespace, but they will not be automatically resent to the originating host. This option might be necessary for legacy iWay applications that handled cookies manually.

The cookie store is managed in memory. By default, the cookies are lost when the server halts. To preserve the contents of the cookie store, set the Persistent Cookie Store parameter to *true*. When persistence is enabled, the cookie store is saved to disk every time it changes. The next time the server reboots, it will reload the cookies from the file. The persistent cookie store files are specific to a server configuration and do not interfere with other configurations.

The provider periodically discards cookies that have expired. This avoids a memory leak and limits the growth of the file on disk. Expired cookies are also discarded when reloading a cookie store after a restart.

The HTTP Nonblocking Emit Service (com.ibi.agents.XDNHttpEmitAgent) is used to send HTTP requests. The HTTP Client Provider parameter for this agent determines which provider will send the request. This indirectly selects the cookie store and therefore the cookies that will be resubmitted to the destination server.

When two HTTP Nonblocking Emit Services must keep their cookies independent, they can use two different Pooling providers. To avoid the proliferation of Pooling providers, it is also possible to name a specific cookie store using the Cookie Store Name parameter for the service. The name could be any identifier chosen by the application. The Pooling provider keeps each named cookie store independent. Just like the default cookie store, a named cookie store is created and managed automatically. If the Pooling provider has the Persistent Cookie Store parameter to *true*, then all named cookie stores are also persistent.

The application might need to create many named cookie stores where each one is needed only for a very short time. This can cause an apparent memory leak, especially if the cookie expiration times are far in the future. The application can control the lifetime of a cookie store by explicitly deleting it with the help of the XDCookieStoreAgent. Set the Action parameter to *Delete Cookie Store* and specify values for the HTTP Client Provider and Cookie Store Name parameters. This will destroy the cookie store and delete the persistent file if applicable. Leave the Cookie Store Name parameter empty to delete the default cookie store. Unlike named cookie stores, the default cookie store is emptied but never destroyed.

Authentication Realms

iWay Service Manager supports pervasive use of authorization realms. Authorization Realms are used to associate Users with Roles and to control user access to resources. Realms are established by logon activities. For example, an HTTP authentication and authorization will establish a realm for the handling of the arriving traffic. The server supports establishment of named realms, each of which can be associated with a login-type operation by its name. The realm associates a user name with the associated credentials to authenticate the user, and to associate that user with named realm tokens.

Available realm types include:

- ❑ **Properties Realm,** which authorizes through a standard properties file. This realm is very simple to manage and is usually associated with demonstration systems or testing.

  The properties file holds <username>=<password> (in clear) and <username>**.role**<n>=rolename

  For example, if *sheila* is a user with password *amanda*, and two roles, her file entries could be:

  ```
  sheila=amanda
  sheila.role0=superuser
  sheila.role1=admin
  ```

- ❑ **Propsrealm**
- ❑ **LDAP Realm,** which authorizes through an LDAP directory. The LDAP is accessed through a previously defined LDAP Provider.
- ❑ **JDBC Realm,** which authorizes through a relational database. The database is accessed through a context provider, which can be a previously defined Data Provider.
- ❑ **Console Realm,** which authorizes through the identities defined for console user access.
- ❑ **JAAS Realm,** which authorizes through a standard Java Authentication and Authorization Service (JAAS). JAAS is an abstraction layer between the server and disparate underlying authentication and authorization mechanisms. Many commercial systems implement the JAAS interface; all documentation for use and setup are obtained through the various commercial packages that implement JAAS.
- ❑ **Adrealm**
- ❑ **Kerberos**

The authorization name and credential, along with the roles associated with that user under that credential, define a principal which is the standard name for this identity. The current principal, including the name and credential (usually a password) and the associated roles can be examined through the iWay Functional Language (iFL) expressions. For example, a process flow can take appropriate branches depending upon whether or not the principal supports a specific role.

Data Quality Providers

iWay Data Quality Server (DQS) is an essential tool for complex data quality management. iWay DQS is designed not only to evaluate, monitor, and manage data quality in different information systems, but also to prevent incorrect data from entering these systems in the first place. iWay DQS is bundled with a specific set of business rules and localized dictionaries.

iWay Data Quality Server (DQS) is integrated with iWay Service Manager (iSM) through the Data Quality provider. There are two Data Quality providers that can be configured:

- **Data Quality Runtime provider**

  This provider represents the DQS runtime, which maintains execution threads for DQS plans invoked by DQS plan services in iSM.

  The Data Quality Runtime provider, introduced in iWay release 7.0, uses one provider for a DQS implementation, and is the preferred provider. New process flow services using this provider take advantage of the pooling capabilities of the provider and offer new services, such as dynamic plan selection, dynamic row construction, and multiple-row input and output.

- **Data Quality provider (deprecated)**

  This provider is used to reference a DQS plan and to manage a pool of execution threads if required. This provider is also bound to a single DQS plan and does not support advanced services, such as *n-row results*.

All iSM components invoke DQS using a defined Data Quality provider. In addition, you can configure one or more providers, each representing a specific DQS plan.

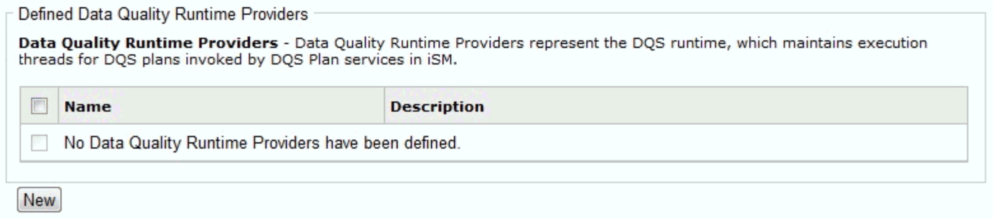
For more information about using iWay DQS, see the *iWay Data Quality Server Getting Started* documentation and *iWay Data Quality Server User's Guide*.

Procedure: How to Define Data Quality Runtime Providers

To define a Data Quality Runtime providers:

1. In the left console pane of the Server menu, select *Data Quality Providers*.

The Data Quality Providers pane opens, as shown in the following image.



Defined Data Quality Runtime Providers

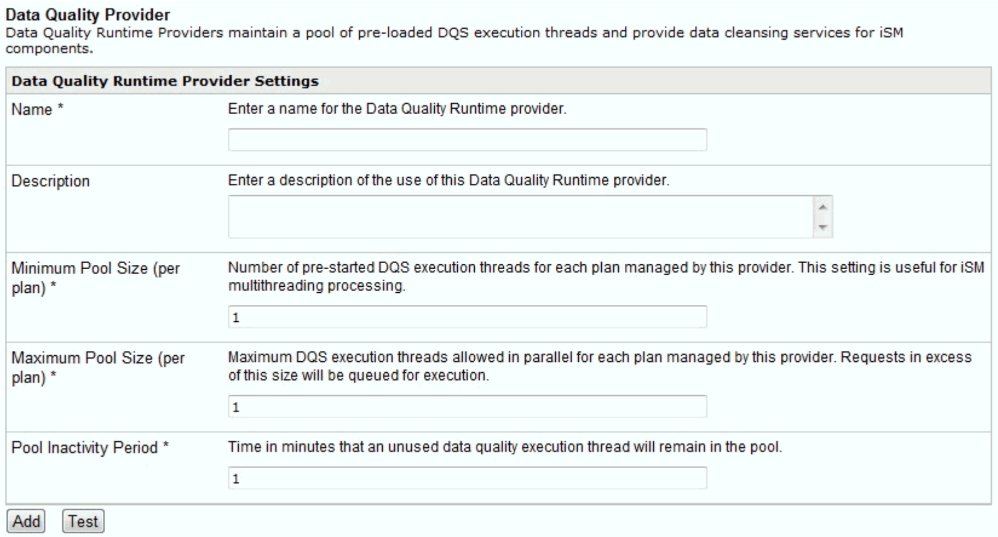
Data Quality Runtime Providers - Data Quality Runtime Providers represent the DQS runtime, which maintains execution threads for DQS plans invoked by DQS Plan services in iSM.

□	Name	Description
□	No Data Quality Runtime Providers have been defined.	

New

2. In the Data Quality Runtime Providers area, click *New*.

   The Configuration pane opens, which contains parameters and settings for the Data Quality Runtime provider, as shown in the following image.



Data Quality Provider

Data Quality Runtime Providers maintain a pool of pre-loaded DQS execution threads and provide data cleansing services for iSM components.

Data Quality Runtime Provider Settings	
Name *	Enter a name for the Data Quality Runtime provider.
Description	Enter a description of the use of this Data Quality Runtime provider.
Minimum Pool Size (per plan) *	Number of pre-started DQS execution threads for each plan managed by this provider. This setting is useful for iSM multithreading processing. 1
Maximum Pool Size (per plan) *	Maximum DQS execution threads allowed in parallel for each plan managed by this provider. Requests in excess of this size will be queued for execution. 1
Pool Inactivity Period *	Time in minutes that an unused data quality execution thread will remain in the pool. 1

Add Test

3. Enter the appropriate values for the Data Quality Runtime provider parameters.
4. Click *Add* when you are finished.

   You are returned to the Data Quality Providers pane, where the new Data Quality Runtime provider that was defined is added to the list.

5. To define multiple Data Quality Runtime provider, repeat this procedure.

Reference: Data Quality Runtime Providers Parameters

The following table lists and describes all of the available parameters and settings for the Data Quality Runtime provider.

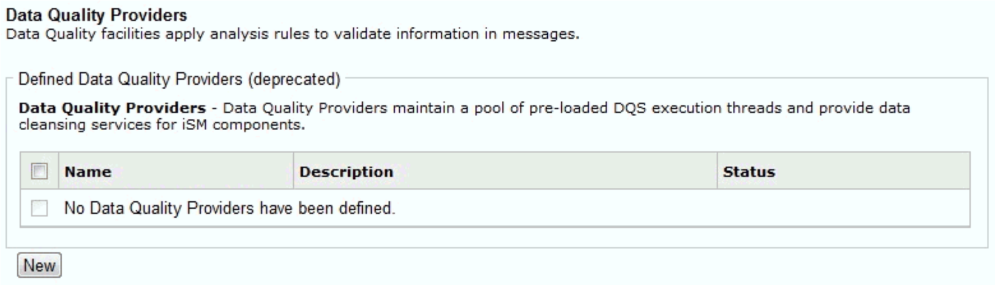
Parameter	Description
Name *	Enter a name for the Data Quality Runtime provider.
Description	Enter a brief description for this Data Quality Runtime provider.
Minimum Pool Size (per plan) *	Number of pre-started DQS execution threads for each plan managed by this provider. This setting is useful for iSM multithreading processing.
Maximum Pool Size (per plan) *	Maximum DQS execution threads allowed in parallel for each plan managed by this provider. Requests in excess of this size will be queued for execution.
Pool Inactivity Period *	Time in minutes that an unused data quality execution thread will remain in the pool.

Procedure: How to Define Data Quality Providers (Deprecated)

To define a Data Quality provider (deprecated):

1. In the left console pane of the Server menu, select *Data Quality Providers*.

   The Data Quality Providers pane opens, as shown in the following image.

   Data Quality Providers
   Data Quality facilities apply analysis rules to validate information in messages.

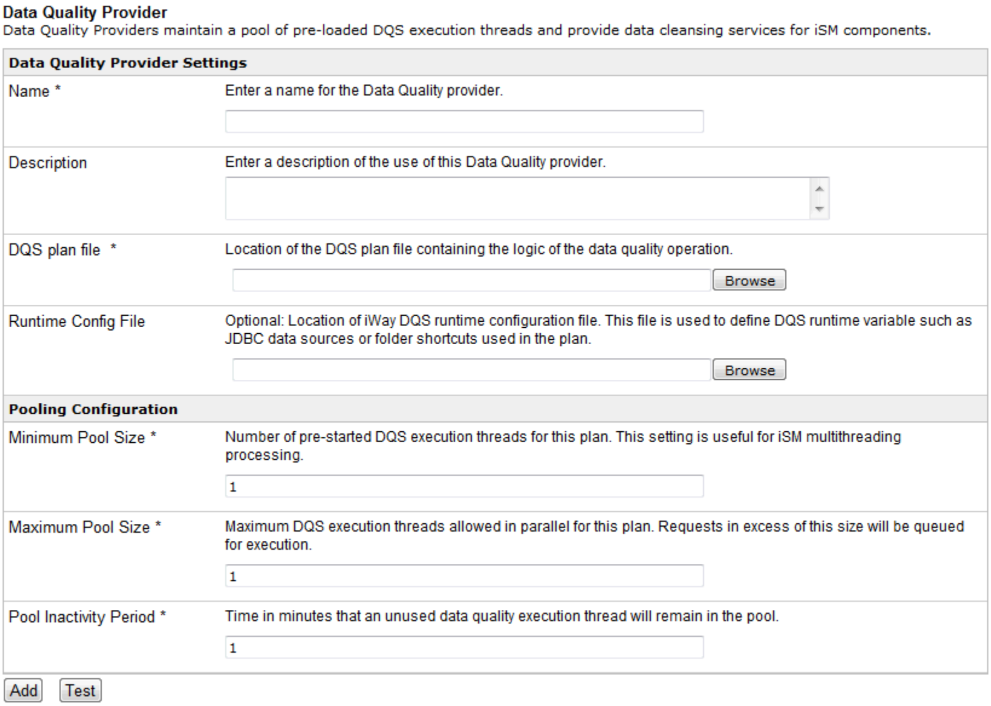
   Defined Data Quality Providers (deprecated)

   Data Quality Providers - Data Quality Providers maintain a pool of pre-loaded DQS execution threads and provide data cleansing services for iSM components.

□	Name	Description	Status
□	No Data Quality Providers have been defined.		

2. In the Data Quality Providers (deprecated) area, click *New*.

The Configuration pane opens, which contains parameters and settings for the Data Quality provider (deprecated), as shown in the following image.



Data Quality Provider

Data Quality Providers maintain a pool of pre-loaded DQS execution threads and provide data cleansing services for iSM components.

Data Quality Provider Settings	
Name *	Enter a name for the Data Quality provider.
Description	Enter a description of the use of this Data Quality provider.
DQS plan file *	Location of the DQS plan file containing the logic of the data quality operation.
Runtime Config File	Optional: Location of iWay DQS runtime configuration file. This file is used to define DQS runtime variable such as JDBC data sources or folder shortcuts used in the plan.
Pooling Configuration	
Minimum Pool Size *	Number of pre-started DQS execution threads for this plan. This setting is useful for iSM multithreading processing. 1
Maximum Pool Size *	Maximum DQS execution threads allowed in parallel for this plan. Requests in excess of this size will be queued for execution. 1
Pool Inactivity Period *	Time in minutes that an unused data quality execution thread will remain in the pool. 1

3. Enter the appropriate values for the Data Quality provider (deprecated) parameters.
4. Click *Add* when you are finished.

   You are returned to the Data Quality Providers pane, where the new Data Quality provider (deprecated) that was defined is added to the list.
5. To define multiple Data Quality providers (deprecated), repeat this procedure.

Reference: Data Quality Providers (Deprecated) Parameters

The following table lists and describes all of the available parameters and settings for the Data Quality provider (deprecated).

Parameter	Description
Data Quality Provider Settings	

Parameter	Description
Name *	Enter a name for the Data Quality provider (deprecated).
Description	Enter a brief description for this Data Quality provider (deprecated).
DQS Plan File *	Location of the DQS plan file containing the logic of the data quality operation.
Runtime Config File	Location of iWay DQS runtime configuration file (optional). This file is used to define DQS runtime variables, such as JDBC data sources or folder shortcuts used in the plan.
Pooling Configuration	
Minimum Pool Size *	Number of pre-started DQS execution threads for this plan. This setting is useful for iSM multithreading processing.
Maximum Pool Size *	Maximum DQS execution threads allowed in parallel for this plan. Requests in excess of this size will be queued for execution.
Pool Inactivity Period *	Time in minutes that an unused data quality execution thread will remain in the pool.

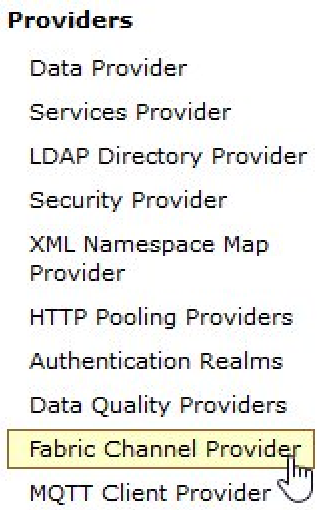
Fabric Channel Provider

A Fabric Channel Provider initializes a Fabric Channel that can execute transactions or listen for events in a Hyperledger Fabric network.

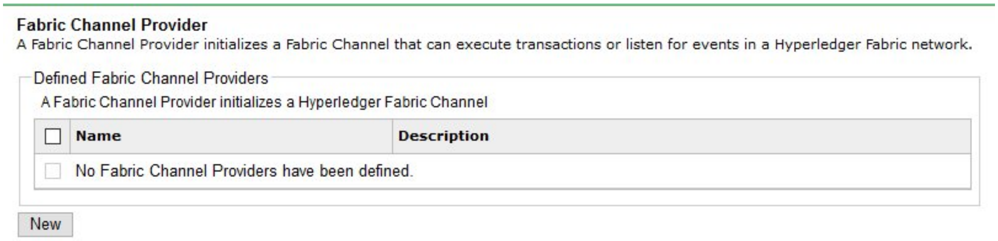
Procedure: How to Configure the Fabric Channel Provider

To configure the Fabric Channel provider:

1. In the left console pane of the Server menu, select *Fabric Channel Provider*.



The Fabric Channel Provider pane opens.



Fabric Channel Provider

A Fabric Channel Provider initializes a Fabric Channel that can execute transactions or listen for events in a Hyperledger Fabric network.

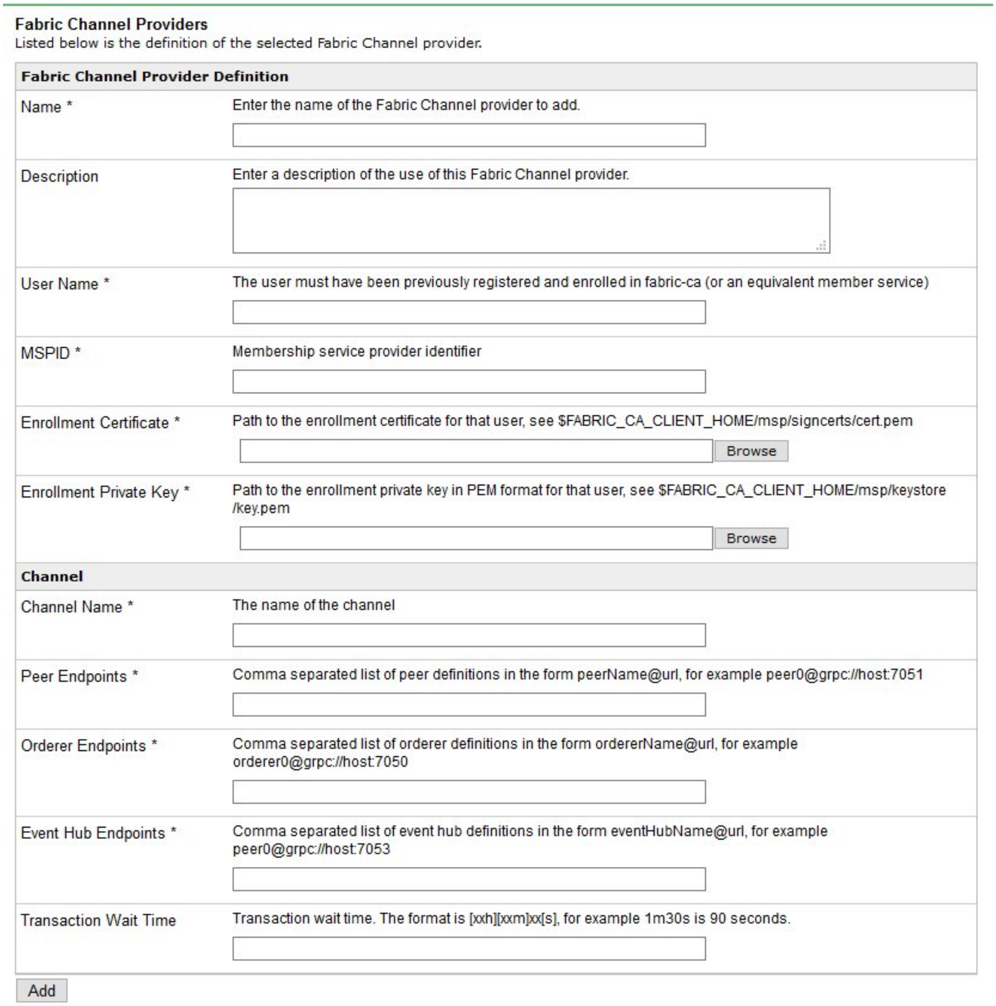
Defined Fabric Channel Providers

A Fabric Channel Provider initializes a Hyperledger Fabric Channel

□	Name	Description
□	No Fabric Channel Providers have been defined.	

2. Click *New* in the Defined Fabric Channel Providers section.

The Fabric Channel Provider Definition pane opens.



Fabric Channel Providers
Listed below is the definition of the selected Fabric Channel provider.

Fabric Channel Provider Definition	
Name *	Enter the name of the Fabric Channel provider to add.
Description	Enter a description of the use of this Fabric Channel provider.
User Name *	The user must have been previously registered and enrolled in fabric-ca (or an equivalent member service)
MSPID *	Membership service provider identifier
Enrollment Certificate *	Path to the enrollment certificate for that user, see $FABRIC_CA_CLIENT_HOME/msp/signcerts/cert.pem
Enrollment Private Key *	Path to the enrollment private key in PEM format for that user, see $FABRIC_CA_CLIENT_HOME/msp/keystore /key.pem
Channel	
Channel Name *	The name of the channel
Peer Endpoints *	Comma separated list of peer definitions in the form peerName@url, for example peer0@grpc://host:7051
Orderer Endpoints *	Comma separated list of orderer definitions in the form ordererName@url, for example orderer0@grpc://host:7050
Event Hub Endpoints *	Comma separated list of event hub definitions in the form eventHubName@url, for example peer0@grpc://host:7053
Transaction Wait Time	Transaction wait time. The format is [xxh][xxm]xx[s], for example 1m30s is 90 seconds.

3. Provide the appropriate values for your Fabric Channel provider parameters as listed and defined in the following tables.

Fabric Channel Provider Definition

Parameter	Description
Name *	Enter the name of the Fabric Channel provider to add.

Parameter	Description
Description	Enter a description of the use of this Fabric Channel provider.
User Name *	The user must have been previously registered and enrolled in fabric-ca (or an equivalent member service).
MSPID *	The membership service provider identifier.
Enrollment Certificate *	The path to the enrollment certificate for that user, see $FABRIC_CA_CLIENT_HOME/msp/signcerts/cert.pem.
Enrollment Private Key *	the path to the enrollment private key in PEM format for that user, see $FABRIC_CA_CLIENT_HOME/msp/keystore/key.pem

Channel

Parameter	Description
Channel Name *	The name of the channel.
Peer Endpoints *	A comma-separated list of peer definitions in the form peerName@url, for example peer0@grpc://host:7051.
Orderer Endpoints *	A comma-separated list of orderer definitions in the form ordererName@url, for example orderer0@grpc://host:7050.
Event Hub Endpoints *	A comma-separated list of event hub definitions in the form eventHubName@url, for example peer0@grpc://host:7053.
Transaction Wait Time	The transaction wait time. The format is [xxh][xxm]xx[s], for example 1m30s is 90 seconds.

4. Click *Add* when you are finished.

   You are returned to the main Fabric Channel Provider pane and the new Fabric Channel provider that was defined is added to the list.

5. To define multiple Fabric Channel providers, repeat this procedure.

MQTT Client Provider

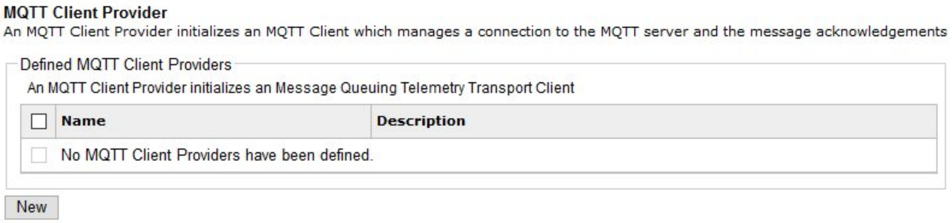
An MQTT Client Provider initializes an MQTT Client which manages a connection to the MQTT server and the message acknowledgements .

Procedure: How to Configure an MQTT Client Provider

To define an MQTT Client provider:

1. In the left console pane of the Server menu, select *MQTT Client Provider*.

   The MQTT Client Provider pane opens.



MQTT Client Provider

An MQTT Client Provider initializes an MQTT Client which manages a connection to the MQTT server and the message acknowledgements

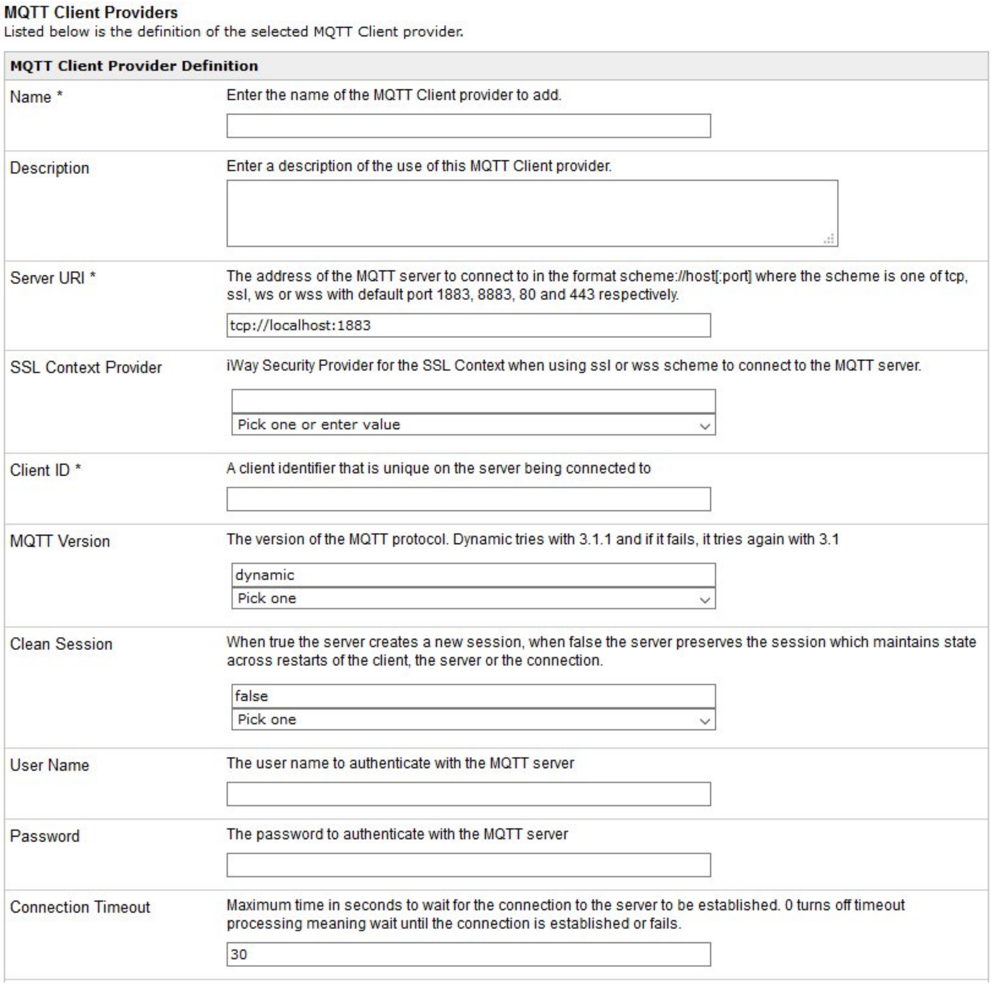
Defined MQTT Client Providers

An MQTT Client Provider initializes an Message Queuing Telemetry Transport Client

□	Name	Description
□	No MQTT Client Providers have been defined.	

2. Click *New* in the Defined MQTT Client Providers section.

The MQTT Client Providers pane opens.



MQTT Client Providers

Listed below is the definition of the selected MQTT Client provider.

MQTT Client Provider Definition

Name *

Enter the name of the MQTT Client provider to add.

Description

Enter a description of the use of this MQTT Client provider.

Server URI *

The address of the MQTT server to connect to in the format scheme://host[:port] where the scheme is one of tcp, ssl, ws or wss with default port 1883, 8883, 80 and 443 respectively.

tcp://localhost:1883

SSL Context Provider

iWay Security Provider for the SSL Context when using ssl or wss scheme to connect to the MQTT server.

Pick one or enter value

Client ID *

A client identifier that is unique on the server being connected to

MQTT Version

The version of the MQTT protocol. Dynamic tries with 3.1.1 and if it fails, it tries again with 3.1

dynamic

Pick one

Clean Session

When true the server creates a new session, when false the server preserves the session which maintains state across restarts of the client, the server or the connection.

false

Pick one

User Name

The user name to authenticate with the MQTT server

Password

The password to authenticate with the MQTT server

Connection Timeout

Maximum time in seconds to wait for the connection to the server to be established. 0 turns off timeout processing meaning wait until the connection is established or fails.

30

3. Provide the appropriate values for your MQTT Client parameters as listed and defined in the following table.

Parameter	Description
Name *	Enter the name of the MQTT Client provider to add.
Description	Enter a description of the use of this MQTT Client provider.

Parameter	Description
Server URI *	The address of the MQTT server to connect to in the format scheme://host[:port] where the scheme is one of tcp, ssl, ws or wss with default port 1883, 8883, 80 and 443 respectively.
SSL Context Provider	iWay Security Provider for the SSL Context when using ssl or wss scheme to connect to the MQTT server.
Client ID *	A client identifier that is unique on the server being connected to.
MQTT Version	The version of the MQTT protocol. Dynamic tries with 3.1.1 and if it fails, it tries again with 3.1.
Clean Session	If set to true, the server creates a new session. If set to false, the server preserves the session, which maintains state across restarts of the client, the server or the connection.
User Name	The user name to authenticate with the MQTT server.
Password	The password to authenticate with the MQTT server.
Connection Timeout	The maximum time in seconds to wait for the connection to the server to be established. 0 turns off timeout processing, meaning wait until the connection is established or fails.
Maximum Inflight	The maximum number of messages that can be sent or received with pending acknowledgments.
Keep Alive Interval	The interval in seconds before sending a keep alive ping message if no other messages are transmitted. The value 0 disables the keep alive processing.
Persistence Store	The persistence store keeps messages until they are acknowledged for *At Least Once* and *Exactly Once* quality of service. The file store is preserved when the provider is recreated (for example at server restart) but the memory store is lost.

4. Click *Add* when you are finished.

   You are returned to the MQTT Client Provider pane and the new MQTT Client provider that was defined is added to the list.

5. To define multiple MQTT Client providers, repeat this procedure.

TCP Connection Provider

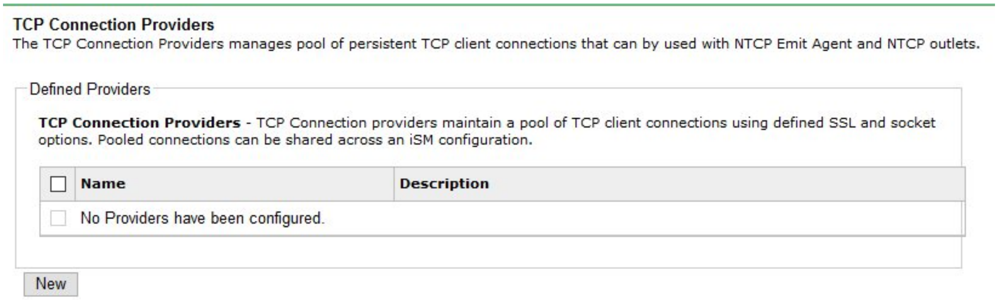
The TCP Connection Providers manages pool of persistent TCP client connections that can by used with NTCP Emit Agent and NTCP outlets.

Procedure: How to Configure a TCP Connection Provider

To configure a TCP Connection Provider:

1. In the left console pane of the Server menu, select *TCP Connection* .

   The TCP Connection Provider Definition pane displays, as shown in the following image.



TCP Connection Providers

The TCP Connection Providers manages pool of persistent TCP client connections that can by used with NTCP Emit Agent and NTCP outlets.

Defined Providers

TCP Connection Providers - TCP Connection providers maintain a pool of TCP client connections using defined SSL and socket options. Pooled connections can be shared across an iSM configuration.

□	Name	Description
□	No Providers have been configured.	

2. Click *New*.
3. Provide the appropriate values for your TCP Connection as listed and defined in the following table.

Parameter	Description
Name *	Enter a name for the TCP Connection Provider
Description	Enter a description for this provider.
Set TCP No Delay	If true, disables Nagle's Algorithm on the client socket. This will result in faster line turnaround at the expense of an increased number of packets. (Advanced).

Parameter	Description
Reuse Socket Address	If true, allows a socket to be bound even though a previous connection is still in a timeout state. Note that this only applies when the local IP interface is specified. This option may be useful for applications that make a large number of short-lived, non-persistent connections, but should be changed with caution. (Advanced).
Linger-On-Close Timeout	This option disables/enables immediate return from a close() of a TCP Socket. Enabling this option with a non-zero Integer timeout means that a close() will block pending the transmission and acknowledgement of all data written to the peer, at which point the socket is closed gracefully. Upon reaching the linger timeout, the socket is closed forcefully, with a TCP RST. Enabling the option with a timeout of zero does a forceful close immediately. Note that this may have the effect of leaving sockets on the server side in a wait state. Enter -1 or leave blank for the JRE default.
Secure Connection	If true, secure the connection using SSL.
SSL Context Provider	The iWay Security Provider for SSL Context. If the SSL Context Provider is left blank, the default provider will be used.
IP Interface Host	The local IP Interface from which the outgoing IP socket originates. (Advanced).
Connection Timeout *	The maximum time in milliseconds to wait for a new connection to the server to be established.

Pooling

Parameter	Description
Maximum Connections Per Address *	Maximum number of connections to cache per address for this provider. Value must be greater than 0.
Maximum Total Connections *	Maximum number of connections to be cached by this provider. Value must be greater than 0.

Idle Connection Eviction

Parameter	Description
Eviction Interval	Time between runs of the idle connection eviction thread, in seconds. A negative value means the eviction thread will never run.
Idle Connection Timeout	The minimum amount of time a connection can remain idle in the pool before the eviction thread disposes of it. When the eviction thread runs and finds that a connection has been idle for at least this interval, the connection will be removed from the pool and closed. Note that a connection can remain in the pool longer than the timeout, depending on the scheduling of the eviction interval.

4. Click *Test* to check for proper connections.
5. To define multiple TCP Connection providers, repeat this procedure.

Token Store Provider

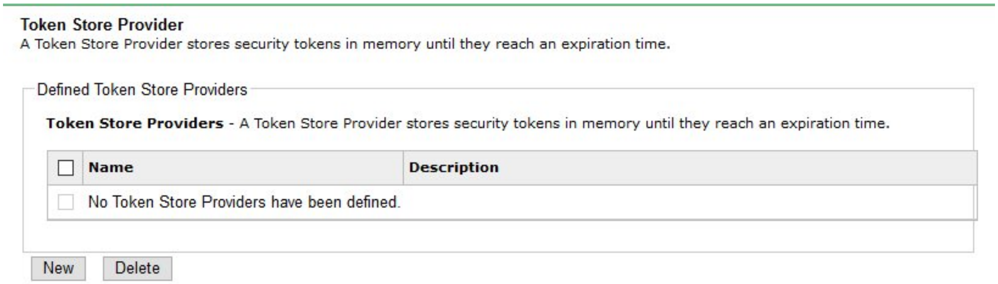
A Token Store Provider stores security tokens in memory until they reach an expiration time.

Procedure: How to Configure a Token Store Provider

To define a Token Store provider using the iWay Service Manager Administration Console:

1. In the left console pane of the Server menu, select *Token Store Provider*.

   The Token Store Provider pane opens.



Token Store Provider
A Token Store Provider stores security tokens in memory until they reach an expiration time.

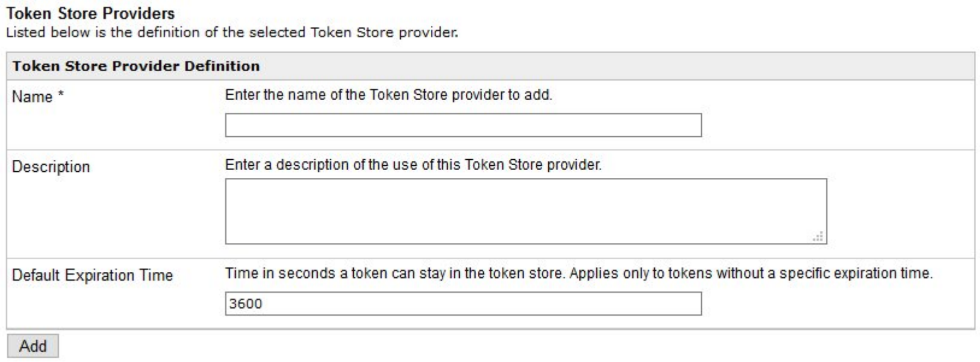
Defined Token Store Providers

Token Store Providers - A Token Store Provider stores security tokens in memory until they reach an expiration time.

□	Name	Description
□	No Token Store Providers have been defined.	

2. Click *New*.

The Token Store Definition pane opens.



Token Store Providers

Listed below is the definition of the selected Token Store provider.

Token Store Provider Definition	
Name *	Enter the name of the Token Store provider to add.
Description	Enter a description of the use of this Token Store provider.
Default Expiration Time	Time in seconds a token can stay in the token store. Applies only to tokens without a specific expiration time. 3600

3. Enter the following parameters for the Token Store provider :

Token Store Provider Definition

Parameter	Description
Name *	Enter the name of the Token Store provider to add.
Description	Enter a description of the use of this Token Store provider.
Default Expiration Time	The time in seconds a token can stay in the token store. Applies only to tokens without a specific expiration time.

4. Click *Add*.

   You are returned to the Token Store Provider pane and the new Token Store provider that was defined is added to the list.

5. To define multiple Token Store providers, repeat this procedure.

Schedule Provider

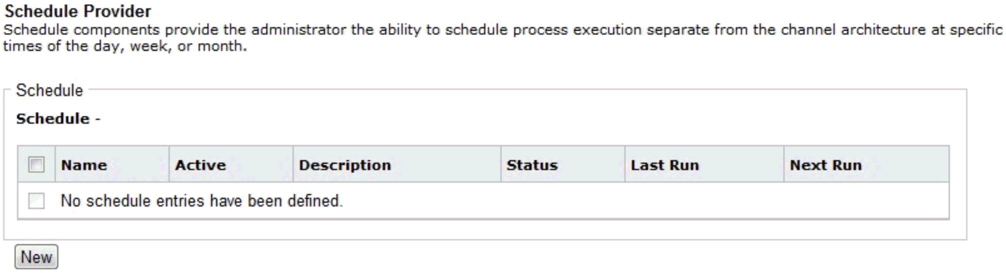
This section describes how to configure Schedule providers with iWay Service Manager (iSM). Schedule providers allow administrators to create a task and schedule it for execution by iSM.

Procedure: How to Configure a Schedule Provider

To configure a Schedule provider:

1. In the left console pane of the Server menu, select *Schedule Provider*.

The Schedule pane opens, as shown in the following image.



Schedule Provider

Schedule components provide the administrator the ability to schedule process execution separate from the channel architecture at specific times of the day, week, or month.

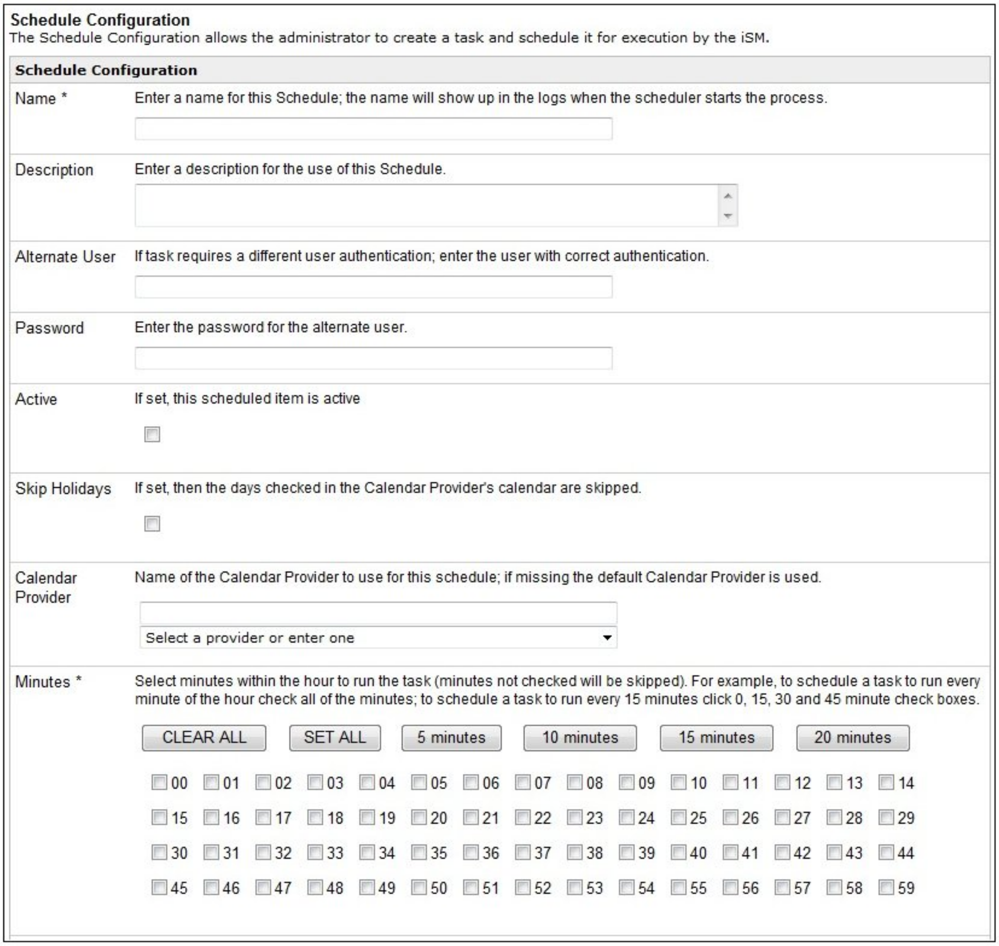
Schedule

Schedule -

□	Name	Active	Description	Status	Last Run	Next Run
□	No schedule entries have been defined.					

2. Click *New*.

The Schedule Configuration pane opens, which contains parameters and settings for the Schedule provider, as shown in the following image.



Schedule Configuration
The Schedule Configuration allows the administrator to create a task and schedule it for execution by the iSM.

Schedule Configuration	
Name *	Enter a name for this Schedule; the name will show up in the logs when the scheduler starts the process.
Description	Enter a description for the use of this Schedule.
Alternate User	If task requires a different user authentication; enter the user with correct authentication.
Password	Enter the password for the alternate user.
Active	If set, this scheduled item is active □
Skip Holidays	If set, then the days checked in the Calendar Provider's calendar are skipped. □
Calendar Provider	Name of the Calendar Provider to use for this schedule; if missing the default Calendar Provider is used.
Minutes *	Select minutes within the hour to run the task (minutes not checked will be skipped). For example, to schedule a task to run every minute of the hour check all of the minutes; to schedule a task to run every 15 minutes click 0, 15, 30 and 45 minute check boxes.

□ 00 □ 01 □ 02 □ 03 □ 04 □ 05 □ 06 □ 07 □ 08 □ 09 □ 10 □ 11 □ 12 □ 13 □ 14
□ 15 □ 16 □ 17 □ 18 □ 19 □ 20 □ 21 □ 22 □ 23 □ 24 □ 25 □ 26 □ 27 □ 28 □ 29
□ 30 □ 31 □ 32 □ 33 □ 34 □ 35 □ 36 □ 37 □ 38 □ 39 □ 40 □ 41 □ 42 □ 43 □ 44
□ 45 □ 46 □ 47 □ 48 □ 49 □ 50 □ 51 □ 52 □ 53 □ 54 □ 55 □ 56 □ 57 □ 58 □ 59

3. Enter the appropriate values for the Schedule provider parameters.
4. Click *Add* when you are finished.

   You are returned to the main Schedule pane, where the new Schedule provider that was defined is added to the list.

5. To define multiple Schedule providers, repeat this procedure.

Reference: Schedule Provider Parameters

The following table lists and describes all of the available parameters for the Schedule provider.

Parameter	Description
Name	Enter a name to identify this Schedule provider. The name will display in the logs when the scheduler starts the process.
Description	Enter a brief description for the use of this Schedule provider.
Alternate User	If the task requires a different user authentication, then enter the user with the correct authentication.
Password	Enter the password for the alternate user.
Active	If set, this scheduled item is active.
Skip Holidays	If checked, iSM will use either the Calendar provider specified for the Calendar Provider parameter or the Calendar provider that is marked as the default provider.
Calendar Provider	Select the defined Calendar provider from the drop-down list that contains the holiday definitions to apply to this schedule. The Calendar provider can be selected from a list of providers configured on this iSM configuration or a Calendar provider from a different configuration can be manually entered. If left blank and the Skip Holidays check box is selected, then the Calendar provider that was selected as the default will be used.
Minutes	Select minutes within the hour to run the task (minutes not checked will be skipped). For example, to schedule a task to run every minute of the hour, check all of the minutes; to schedule a task to run every 15 minutes click 0, 15, 30 and 45 minute check boxes.
Hours	Select hours within the day to run the task (hours not checked will be skipped). For example, to schedule a task to run every hour of the day, check all of the hours in both the AM and PM row.

Parameter	Description
Month	Select the month within the year to run the task (months not checked will be skipped). For example, to schedule a task to run every month check all the months. To schedule a task to only run in January, check the Jan check box.
Weekday	Select day of week to run (days not checked and not implied by Day of Month will be skipped). Weekday and Day of Month are cumulative. For example, every Wednesday AND the 15th of the month.
Day of Month	Select day within the month to run the task (days not checked and not implied by Weekday will be skipped). Day of Week and Day of Month are cumulative. For example, every Wednesday AND the 15th of the month.
Command	iSM command to be scheduled for execution. For more information on a specific command, see BAD XREF HERE "iWay Service Manager Console Commands.
Duration Timer	The length of time that the task will run prior to the Dependent Command. The format of duration [in seconds] is in the following format: `[xxh][xxm]xx[s]` For example, 04h30m45, which creates a duration of 4 hours, 30 minutes, and 45 seconds.
Dependent Command	Command to be executed after the Duration Timer of the task has expired. The command can be directed to any configuration / iWay Integration Application (iIA) by including it in a remote command.

Calendar Provider

The Calendar provider allows you to define 18 months (starting with the current month) of Holiday/Skip dates, which when used with the Schedule provider, allows you to fine tune the scheduler execution.

Once configured, the Calendar provider can be:

1. Exported to a centralized file system.
2. Imported from a centralized file system and used within this iWay Service Manager (iSM) instance.
3. Imported from a centralized Service Manager and used within this iSM instance.
4. Accessed from a centralized iSM.

This allows the Calendar provider to be shared in real time by any iSM that has access to it.

Note: The Calendar Provider **must** be updated every 18 months. Otherwise, once expired, any Schedule providers that use this calendar will execute their tasks as if no Calendar provider was present.

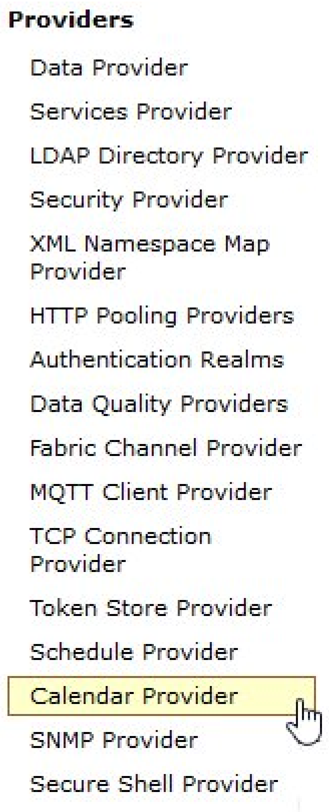
Configuring the Calendar Provider

This section describes all of the steps (including reference information) that are required to configure the Calendar provider.

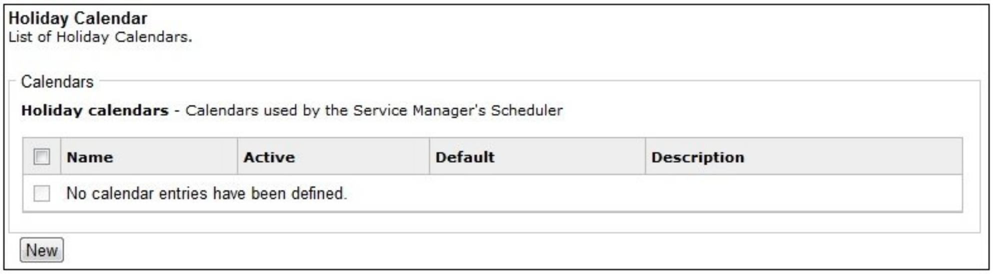
Procedure: How to Configure a Calendar Provider

To configure a Calendar provider:

1. In the left console pane of the Server menu, select *Calendar Provider*, as shown in the following image.



The Holiday Calendar pane opens, as shown in the following image.



Holiday Calendar
List of Holiday Calendars.

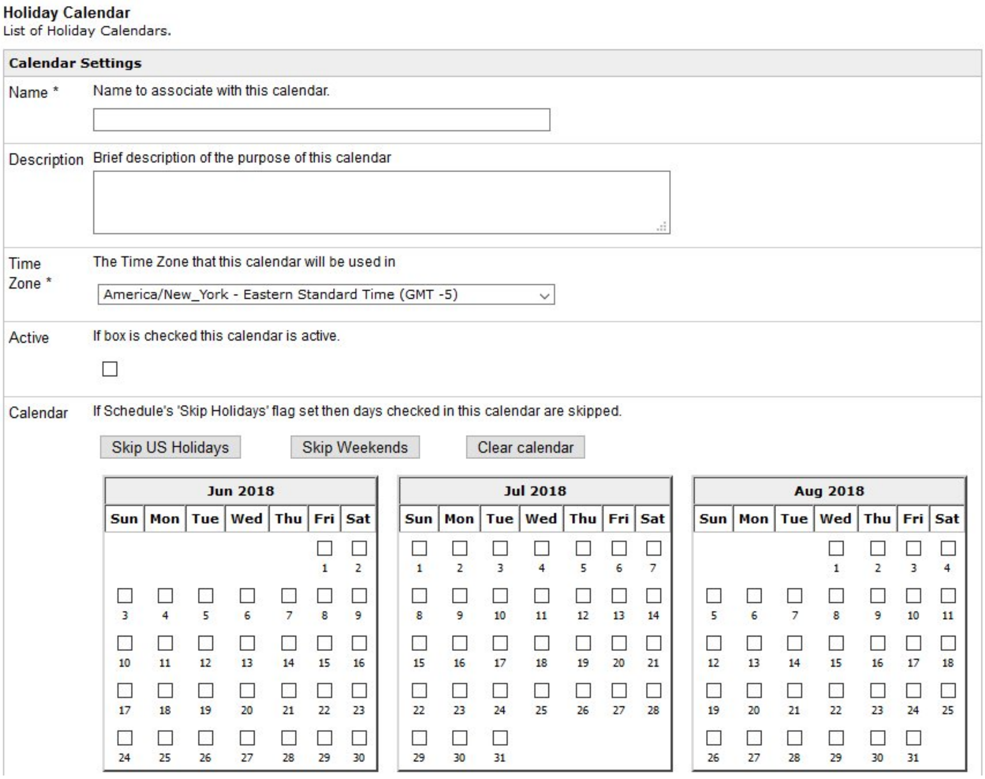
Calendars

Holiday calendars - Calendars used by the Service Manager's Scheduler

□	Name	Active	Default	Description
□	No calendar entries have been defined.			

2. Click *New*.

   The Calendar Settings configuration pane opens, which contains parameters and settings for the Calendar provider, as shown in the following image.



Holiday Calendar
List of Holiday Calendars.

Calendar Settings

Name * — Name to associate with this calendar.

Description — Brief description of the purpose of this calendar

Time Zone * — The Time Zone that this calendar will be used in

America/New_York - Eastern Standard Time (GMT -5)

Active — If box is checked this calendar is active.

□

Calendar — If Schedule's 'Skip Holidays' flag set then days checked in this calendar are skipped.

Jun 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					□ 1	□ 2
□ 3	□ 4	□ 5	□ 6	□ 7	□ 8	□ 9
□ 10	□ 11	□ 12	□ 13	□ 14	□ 15	□ 16
□ 17	□ 18	□ 19	□ 20	□ 21	□ 22	□ 23
□ 24	□ 25	□ 26	□ 27	□ 28	□ 29	□ 30

Jul 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
□ 1	□ 2	□ 3	□ 4	□ 5	□ 6	□ 7
□ 8	□ 9	□ 10	□ 11	□ 12	□ 13	□ 14
□ 15	□ 16	□ 17	□ 18	□ 19	□ 20	□ 21
□ 22	□ 23	□ 24	□ 25	□ 26	□ 27	□ 28
□ 29	□ 30	□ 31				

Aug 2018

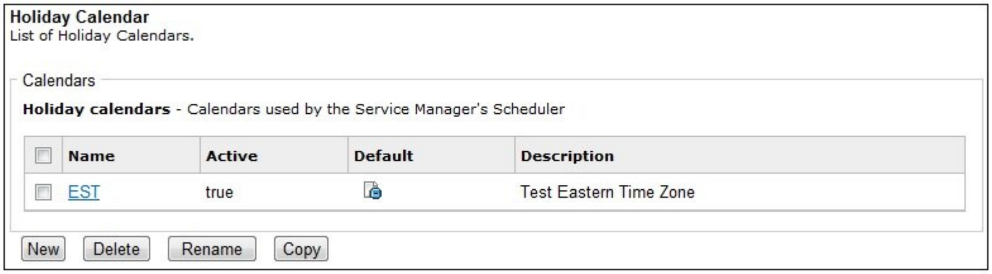
Sun	Mon	Tue	Wed	Thu	Fri	Sat
			□ 1	□ 2	□ 3	□ 4
□ 5	□ 6	□ 7	□ 8	□ 9	□ 10	□ 11
□ 12	□ 13	□ 14	□ 15	□ 16	□ 17	□ 18
□ 19	□ 20	□ 21	□ 22	□ 23	□ 24	□ 25
□ 26	□ 27	□ 28	□ 29	□ 30	□ 31	

3. Enter the appropriate values for the Calendar provider parameters.

   For more information, see *Calendar Provider Parameters*.

4. Scroll down and click *Add* when you are finished.

You are returned to the main Holiday Calendar pane, where the new Calendar provider that was configured is now added to the list. For example:



Holiday Calendar
List of Holiday Calendars.

Calendars

Holiday calendars - Calendars used by the Service Manager's Scheduler

□	Name	Active	Default	Description
□	EST	true		Test Eastern Time Zone

Note: A default Calendar provider may be selected from this list. Click the icon that is in the Default column. Only one default Calendar provider is recognized for each iSM instance.

5. To define multiple Calendar providers, repeat this procedure.

Reference: Calendar Provider Parameters

The following table lists and describes all of the available parameters for the Calendar provider.

Parameter	Description
Name	Enter a name for the Calendar provider. The name must not contain embedded blanks.
Description	Enter a brief description for the Calendar provider. This description will be displayed on the Holiday Calendar pane where available Calendar providers are listed.
Time Zone	Select a specific time zone to use for the Calendar provider from the drop-down list. This allows the scheduler to take into account what time zone relative to the iSM that the schedule's skipping should be applied. This is useful when the iSM is running in one time zone and controlling schedules that run based on a different time zone's values.

Parameter	Description
Active	The Calendar provider may be configured and used by a Schedule provider. However, if the Calendar provider is not marked as active, then the Schedule provider will not use that Calendar provider. To activate the Calendar provider, click the the Active check box. If selected, then this Calendar provider is active.
Calendar	Allows you to select the specific days over the next 18 months to skip. If the check box above a date is selected, then that date is skipped by the Schedule provider, which is configured to use this Calendar provider.

Using the Convenience Buttons

In addition the being able to select specific dates, there are three convenience buttons that will either check specific dates, or uncheck (clear) the dates in the calendar:

- Skip US Holidays
- Skip Weekends
- Clear calendar

The convenience buttons allow you to quickly configure the schedule for the next 18 months.

Skip US Holidays

If you want to set the Calendar provider to skip holidays, the following United States holidays are recognized per year:

- New Year's Day
- Martin Luther King, Jr. Day
- President's Day
- Memorial Day

❑ Independence Day

❑ Labor Day

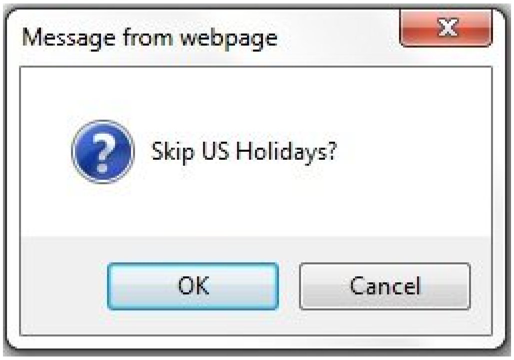
❑ Columbus Day

❑ Thanksgiving Day

❑ Christmas Day

To set those holidays into the current 18 month calendar, click *Skip US Holidays*.

The following confirmation dialog box is displayed:

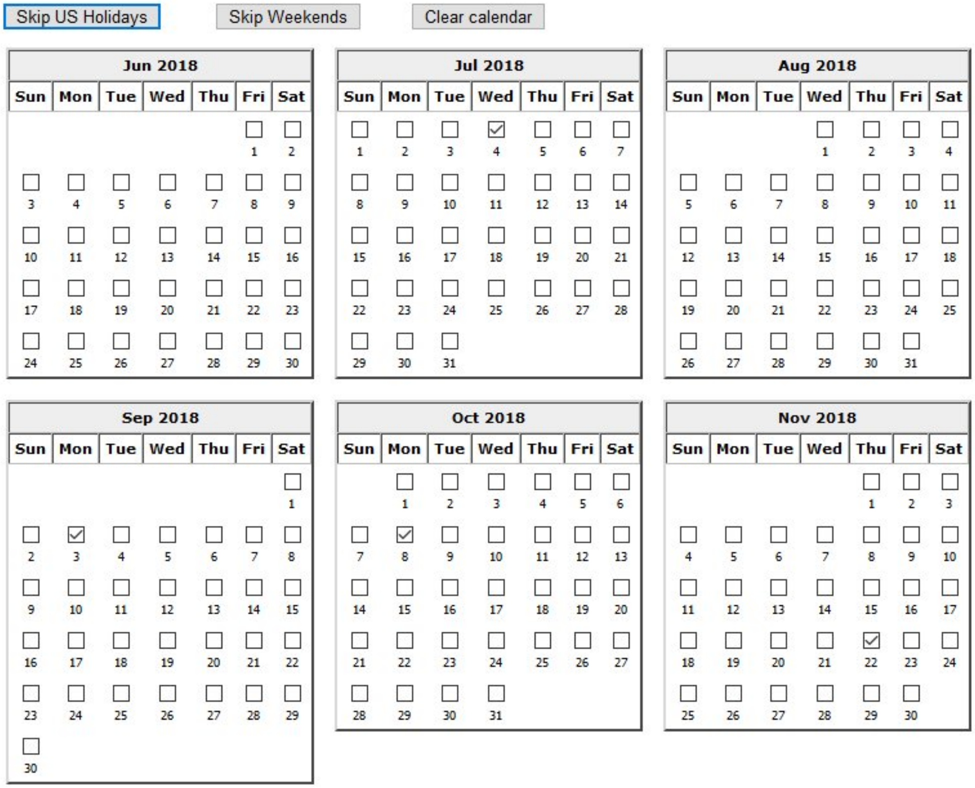


Message from webpage

Skip US Holidays?

To set the holidays on the calendar, click *OK*. Clicking *Cancel* returns you to the Calendar Settings configuration pane without setting any dates.

The following image shows the resulting calendar after the Skip US Holidays button is selected and confirmed:



Jun 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					□ 1	□ 2
□ 3	□ 4	□ 5	□ 6	□ 7	□ 8	□ 9
□ 10	□ 11	□ 12	□ 13	□ 14	□ 15	□ 16
□ 17	□ 18	□ 19	□ 20	□ 21	□ 22	□ 23
□ 24	□ 25	□ 26	□ 27	□ 28	□ 29	□ 30

Jul 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
□ 1	□ 2	□ 3	☑ 4	□ 5	□ 6	□ 7
□ 8	□ 9	□ 10	□ 11	□ 12	□ 13	□ 14
□ 15	□ 16	□ 17	□ 18	□ 19	□ 20	□ 21
□ 22	□ 23	□ 24	□ 25	□ 26	□ 27	□ 28
□ 29	□ 30	□ 31				

Aug 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
			□ 1	□ 2	□ 3	□ 4
□ 5	□ 6	□ 7	□ 8	□ 9	□ 10	□ 11
□ 12	□ 13	□ 14	□ 15	□ 16	□ 17	□ 18
□ 19	□ 20	□ 21	□ 22	□ 23	□ 24	□ 25
□ 26	□ 27	□ 28	□ 29	□ 30	□ 31	

Sep 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						□ 1
□ 2	☑ 3	□ 4	□ 5	□ 6	□ 7	□ 8
□ 9	□ 10	□ 11	□ 12	□ 13	□ 14	□ 15
□ 16	□ 17	□ 18	□ 19	□ 20	□ 21	□ 22
□ 23	□ 24	□ 25	□ 26	□ 27	□ 28	□ 29
□ 30						

Oct 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	□ 1	□ 2	□ 3	□ 4	□ 5	□ 6
□ 7	☑ 8	□ 9	□ 10	□ 11	□ 12	□ 13
□ 14	□ 15	□ 16	□ 17	□ 18	□ 19	□ 20
□ 21	□ 22	□ 23	□ 24	□ 25	□ 26	□ 27
□ 28	□ 29	□ 30	□ 31			

Nov 2018

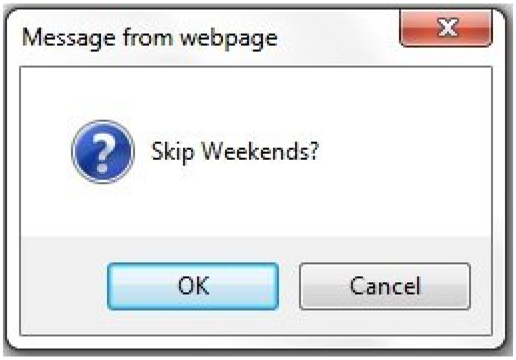
Sun	Mon	Tue	Wed	Thu	Fri	Sat
				□ 1	□ 2	□ 3
□ 4	□ 5	□ 6	□ 7	□ 8	□ 9	□ 10
□ 11	□ 12	□ 13	□ 14	□ 15	□ 16	□ 17
□ 18	□ 19	□ 20	□ 21	☑ 22	□ 23	□ 24
□ 25	□ 26	□ 27	□ 28	□ 29	□ 30	

Skip Weekends

The Calendar provider considers a weekend to be Saturday and Sunday. To set each weekend for the 18 month period is a time consuming task. The Skip Weekends button performs this action with a single click.

To skip all weekends in the current 18 month calendar, click *Skip Weekends*.

The following confirmation dialog box is displayed:

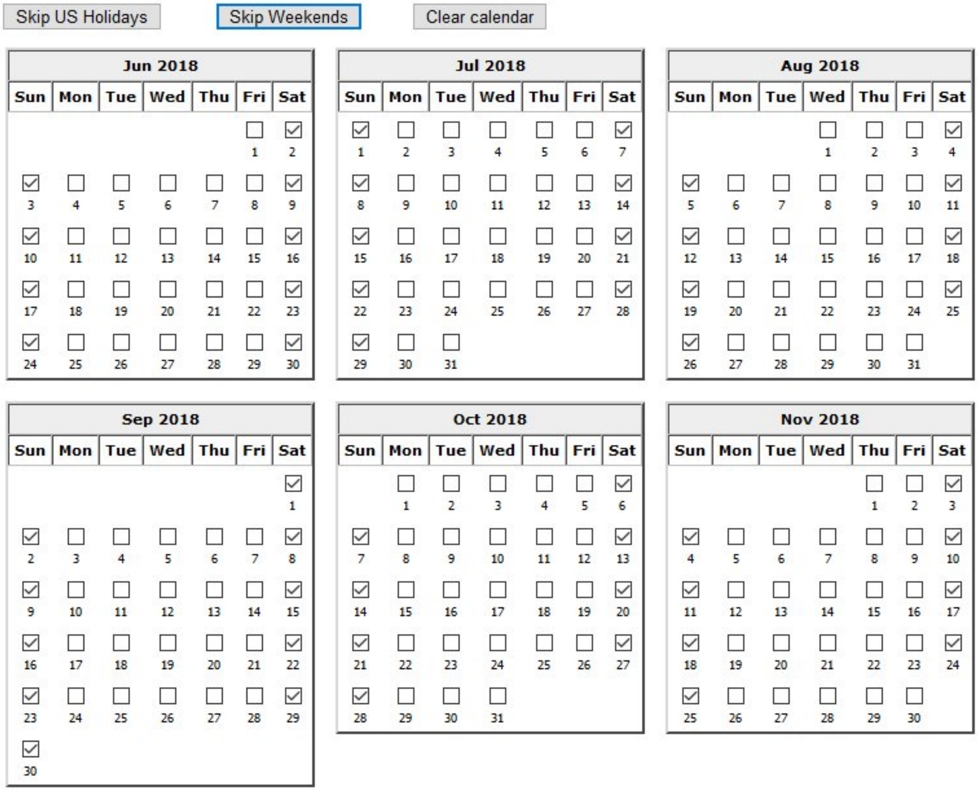


Message from webpage

Skip Weekends?

To skip all weekends on the calendar, click *OK*. Clicking *Cancel* returns you to the Calendar Settings configuration pane without skipping any weekends.

The following image shows the resulting calendar after the Skip Weekends button is selected and confirmed:



Jun 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					☐ 1	☑ 2
☑ 3	☐ 4	☐ 5	☐ 6	☐ 7	☐ 8	☑ 9
☑ 10	☐ 11	☐ 12	☐ 13	☐ 14	☐ 15	☑ 16
☑ 17	☐ 18	☐ 19	☐ 20	☐ 21	☐ 22	☑ 23
☑ 24	☐ 25	☐ 26	☐ 27	☐ 28	☐ 29	☑ 30

Jul 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
☑ 1	☐ 2	☐ 3	☐ 4	☐ 5	☐ 6	☑ 7
☑ 8	☐ 9	☐ 10	☐ 11	☐ 12	☐ 13	☑ 14
☑ 15	☐ 16	☐ 17	☐ 18	☐ 19	☐ 20	☑ 21
☑ 22	☐ 23	☐ 24	☐ 25	☐ 26	☐ 27	☑ 28
☑ 29	☐ 30	☐ 31				

Aug 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
			☐ 1	☐ 2	☐ 3	☑ 4
☑ 5	☐ 6	☐ 7	☐ 8	☐ 9	☐ 10	☑ 11
☑ 12	☐ 13	☐ 14	☐ 15	☐ 16	☐ 17	☑ 18
☑ 19	☐ 20	☐ 21	☐ 22	☐ 23	☐ 24	☑ 25
☑ 26	☐ 27	☐ 28	☐ 29	☐ 30	☐ 31	

Sep 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						☑ 1
☑ 2	☐ 3	☐ 4	☐ 5	☐ 6	☐ 7	☑ 8
☑ 9	☐ 10	☐ 11	☐ 12	☐ 13	☐ 14	☑ 15
☑ 16	☐ 17	☐ 18	☐ 19	☐ 20	☐ 21	☑ 22
☑ 23	☐ 24	☐ 25	☐ 26	☐ 27	☐ 28	☑ 29
☑ 30						

Oct 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5	☑ 6
☑ 7	☐ 8	☐ 9	☐ 10	☐ 11	☐ 12	☑ 13
☑ 14	☐ 15	☐ 16	☐ 17	☐ 18	☐ 19	☑ 20
☑ 21	☐ 22	☐ 23	☐ 24	☐ 25	☐ 26	☑ 27
☑ 28	☐ 29	☐ 30	☐ 31			

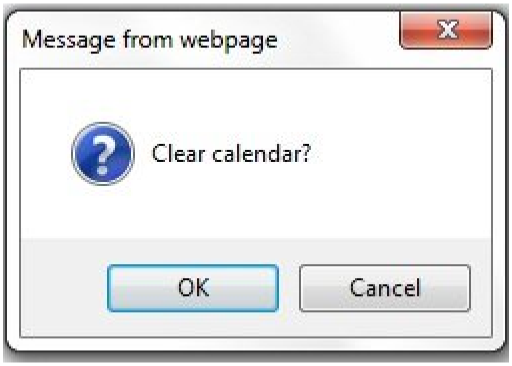
Nov 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				☐ 1	☐ 2	☑ 3
☑ 4	☐ 5	☐ 6	☐ 7	☐ 8	☐ 9	☑ 10
☑ 11	☐ 12	☐ 13	☐ 14	☐ 15	☐ 16	☑ 17
☑ 18	☐ 19	☐ 20	☐ 21	☐ 22	☐ 23	☑ 24
☑ 25	☐ 26	☐ 27	☐ 28	☐ 29	☐ 30	

Clear Calendar

To start over again with a clean calendar and deselect all date check boxes, click *Clear calendar*.

The following confirmation dialog box is displayed:



Message from webpage

Clear calendar?

To clear all of the dates set on the calendar, click *OK*. Clicking *Cancel* returns you to the Calendar Settings configuration pane without resetting any dates.

Using the Function Buttons

The function buttons interface with the current configuration of iSM.

Function Button	Description
Add	Displayed only when adding new Calendar providers. When you are satisfied with the calendar settings clicking Add commits the calendar settings and writes the calendar data to the iSM configuration.
Update	The Update button is displayed in conjunction with the Reload button when the Calendar provider already exists within the iSM configuration. Like the Add button, when you are satisfied with the calendar settings, clicking Update commits the calendar settings and writes the calendar data to the iSM configuration.
Reload	At any time during an update of the Calendar provider, if you want to discard your changes the current Calendar provider may be reloaded from iSM by clicking Reload. All changes to the Calendar provider are discarded and a fresh copy (as of the last Add or Update action) is reloaded from the iSM configuration.

SNMP Provider

SNMP (Simple Network Management Protocol) is a standard Internet protocol to monitor attached devices for conditions or situations that warrant attention. Within a managed network, iWay Service Manager (iSM) is treated as a device to be monitored.

Under SNMP, the two key components are a manager that aggregates, evaluates, and displays information from agents, which represent the managed devices. Typically an iSM user will already have a manager installed for network control. The manager is an external, non-iWay component to which iSM reports. Managers range from simple freeware versions that are downloadable from the web to sophisticated management systems from major vendors.

SNMP exposes management data from the agent in the form of described variables on the managed systems devices. These variables can then be queried (and sometimes set) by the manager.

The SNMP standard does not define which information (variables) a managed device should offer. SNMP uses an extensible design, where the available information is defined by management information bases (MIBs). MIBs describe the structure of the management data of a device subsystem.

When exposing information using SNMP, iSM assumes the role of a managed device. The SNMP provider acts as the SNMP agent. A provided MIB details the information that iSM exposes, including:

- server start and end
- listener start and end
- listener execution statistics including messages processed and their execution time
- special registers (SREGs)

You may have a requirement to monitor iSM along with Java and operating system values. The Java Virtual Machine (JVM) and most operating systems offer MIBs that can also be loaded into your manager. You can then combine these to provide a more complete report of server activity and resource use.

The SNMP remote function calls (RFCs) describe three protocol standards, which are known as V1, V2c, and V3. Managers can choose to implement one or more of these standards. The iSM SNMP provider supports all three standards.

The SNMP facilities are automatically installed during the iSM installation. The MIBs for iSM are located in the following directory:

```
<iwayhome>/etc/mibs
```

The MIBs are copied to the manager software (and usually compiled) as required by that software. The MIBS for Java and the operating system, can be found according to their own software installation. For example, the Java MIB provides access to Java execution information, such as threads, memory use, and semaphores (monitors).

Configuring the SNMP Provider

The iSM SNMP agent is exposed as a provider. There can be one or more providers defined in a configuration, although more than one is only required if there are independent managers controlling different aspects of the server.

To complete the configuration, you will need to know the configuration of the manager. For example, the manager will be configured to interact with the agent on a specific port (usually 161). If you have multiple configurations on the same installation host, you will need to separate these by having different addresses. For performance reasons, iSM uses separate providers on each configuration rather than having a single provider poll other configurations.

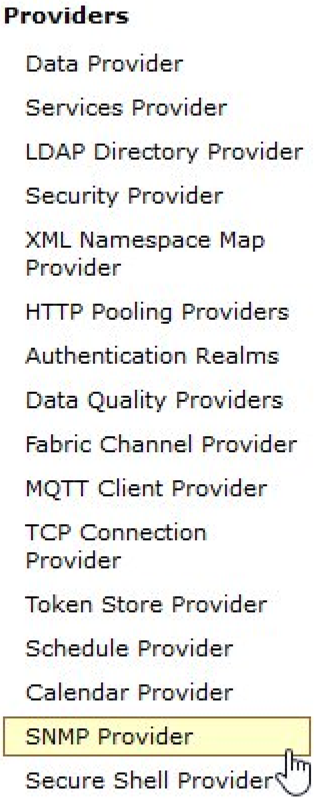
You will also be requested to specify which protocol(s) are used by your manager. Select the set that is supported by the manager.

The provider can accept simultaneous requests from multiple managers, and some managers are capable of sending multiple requests to their agents in parallel. If this is the case, you can specify the number of expected parallel requests by setting the number of execution threads.

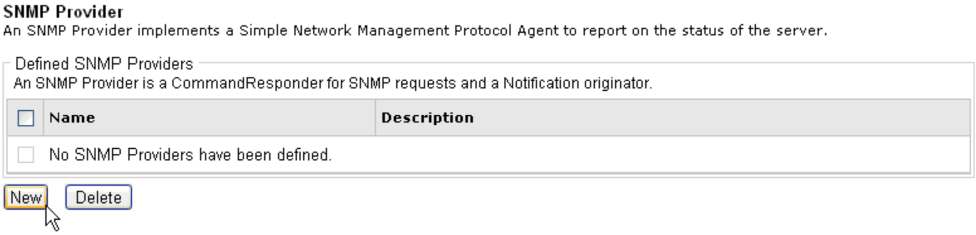
Procedure: How to Configure the SNMP Provider

To configure the SNMP provider:

1. In the left console pane of the Server menu, select *SNMP Provider*.



The SNMP Provider pane opens.



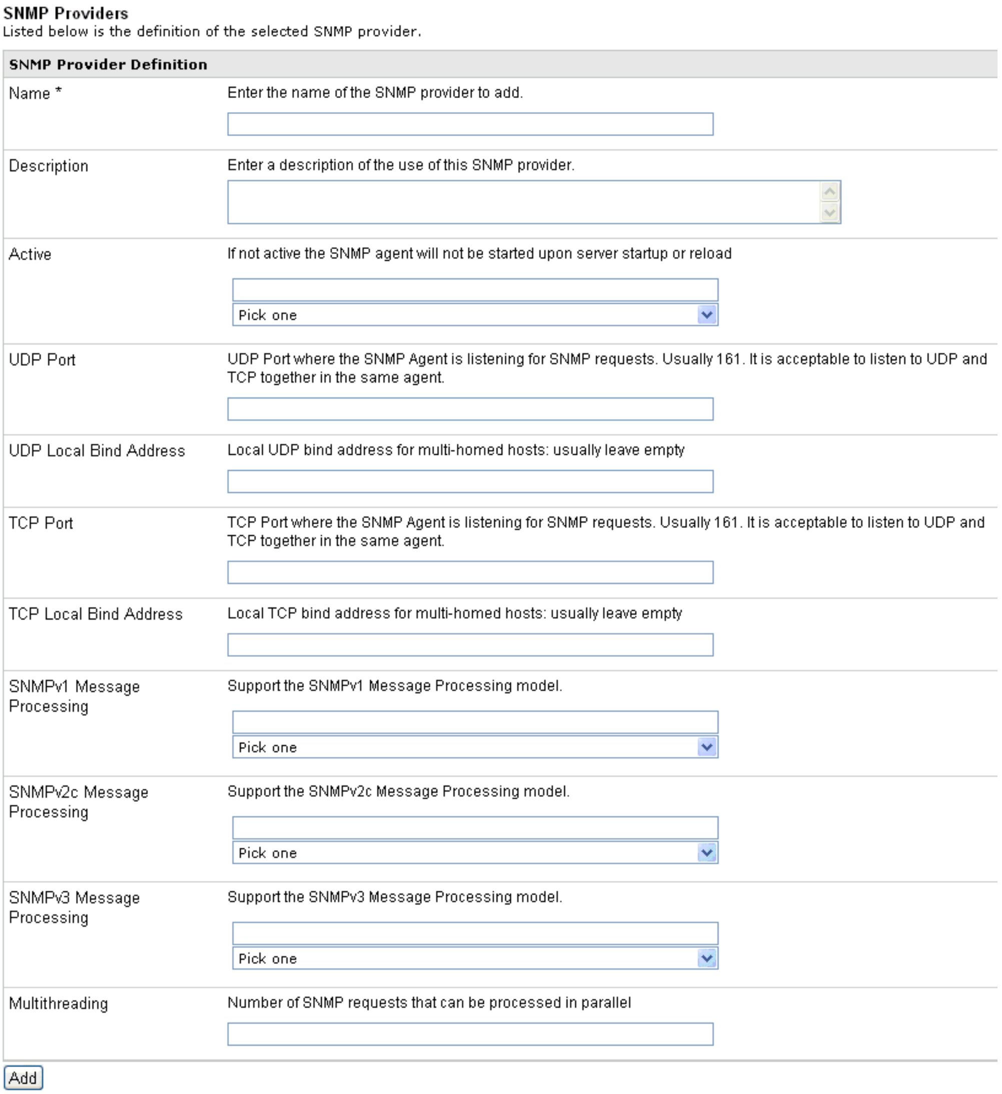
SNMP Provider
An SNMP Provider implements a Simple Network Management Protocol Agent to report on the status of the server.

Defined SNMP Providers
An SNMP Provider is a CommandResponder for SNMP requests and a Notification originator.

□	Name	Description
□	No SNMP Providers have been defined.	

2. Click *New* in the Defined SNMP Providers section.

The SNMP Provider Definition pane opens.



SNMP Providers
Listed below is the definition of the selected SNMP provider.

SNMP Provider Definition	
Name *	Enter the name of the SNMP provider to add.
Description	Enter a description of the use of this SNMP provider.
Active	If not active the SNMP agent will not be started upon server startup or reload Pick one
UDP Port	UDP Port where the SNMP Agent is listening for SNMP requests. Usually 161. It is acceptable to listen to UDP and TCP together in the same agent.
UDP Local Bind Address	Local UDP bind address for multi-homed hosts: usually leave empty
TCP Port	TCP Port where the SNMP Agent is listening for SNMP requests. Usually 161. It is acceptable to listen to UDP and TCP together in the same agent.
TCP Local Bind Address	Local TCP bind address for multi-homed hosts: usually leave empty
SNMPv1 Message Processing	Support the SNMPv1 Message Processing model. Pick one
SNMPv2c Message Processing	Support the SNMPv2c Message Processing model. Pick one
SNMPv3 Message Processing	Support the SNMPv3 Message Processing model. Pick one
Multithreading	Number of SNMP requests that can be processed in parallel

3. Provide the appropriate values for your SNMP provider parameters as listed and defined in the following tables.

Parameter	**Description**
Name *	Enter the name of the SNMP provider to add.

Parameter	Description
Description	Enter a brief description of the of the SNMP provider.
Active	If set to *false*, the SNMP agent will not be started during iSM server startup or restart.
UDP Port	The UDP port where the SNMP agent is listening for SNMP requests. Usually 161. It is acceptable to listen to UDP and TCP together in the same agent.
UDP Local Bind Address	The local UDP bind address for multi-homed hosts. This parameter value is usually left blank.
TCP Port	The TCP port where the SNMP agent is listening for SNMP requests. Usually 161. It is acceptable to listen to UDP and TCP together in the same agent.
TCP Local Bind Address	The local TCP bind address for multi-homed hosts. This parameter value is usually left blank.
SNMPv1 Message Processing	Select *true* to support the SNMPv1 message processing model.
SNMPv2c Message Processing	Select *true* to support the SNMPv2c message processing model.
SNMPv3 Message Processing	Select *true* to support the SNMPv3 message processing model.
Multithreading	The number of SNMP requests that can be processed in parallel.

Community

Parameter	Description
Community	The name of the community for SNMPv1 and SNMPv2c. This acts like a weak password.

User

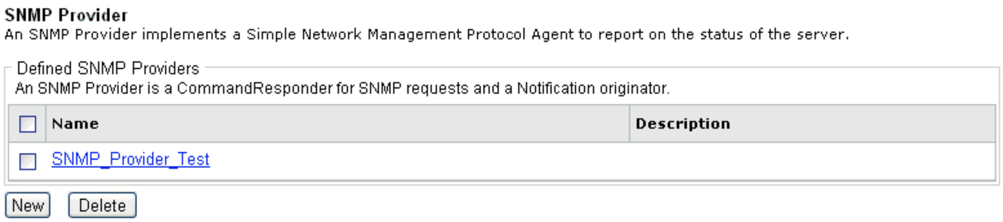
Parameter	Description
User Name	The name of the user to register in the User-based Security Model MIB for SNMPv3.
Authentication Protocol	The authentication protocol to use if authentication is enabled in the security level.
Authentication Passphrase	The authentication passphrase to use if authentication is enabled in the security level.
Privacy Protocol	The privacy protocol to use if privacy is enabled in the security level.
Privacy Passphrase	The privacy passphrase to use if privacy is enabled in the security level.

Notification

Parameter	Description
Send Notifications	Determines whether the server will send SNMP notifications.
Notification Processing Model	The message processing model used to send the notification.
Notification Type	The type of Protocol Data Unit used to send the notification. The Inform notification requires processing model v2c or v3.
Notification Protocol	The protocol over which the notification will be sent.
Notification Host	The host where the notification will be sent.
Notification Port	The port where the notification will be sent. The default is 162.
Notification Timeout	The time allocated to send the notification in 100th of a second.
Notification Retry Count	The number of attempts to send the notification. The default is 1.

4. Click *Add* when you are finished.

   You are returned to the main SNMP Provider pane and the new SNMP provider that was defined is added to the list.



SNMP Provider
An SNMP Provider implements a Simple Network Management Protocol Agent to report on the status of the server.

Defined SNMP Providers
An SNMP Provider is a CommandResponder for SNMP requests and a Notification originator.

□	Name	Description
□	SNMP_Provider_Test	

5. To define multiple SNMP providers, repeat this procedure.

Secure Shell Providers

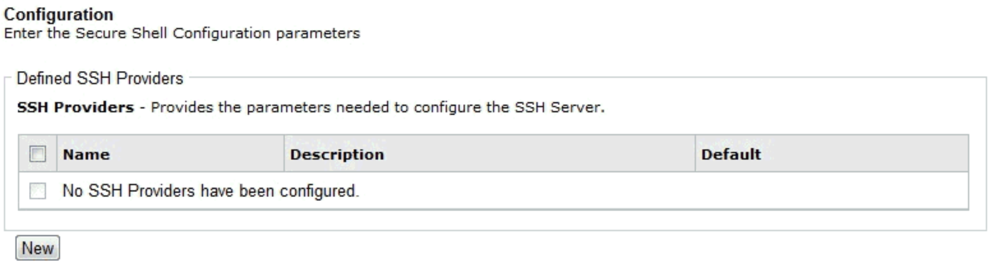
This section describes how to configure Secure Shell (SSH) providers with iWay Service Manager (iSM).

Procedure: How to Define Secure Shell Providers

To define a Secure Shell (SSH) provider:

1. In the left console pane of the Server menu, select *Secure Shell Provider*.

   The Defined SSH Providers pane opens, as shown in the following image.



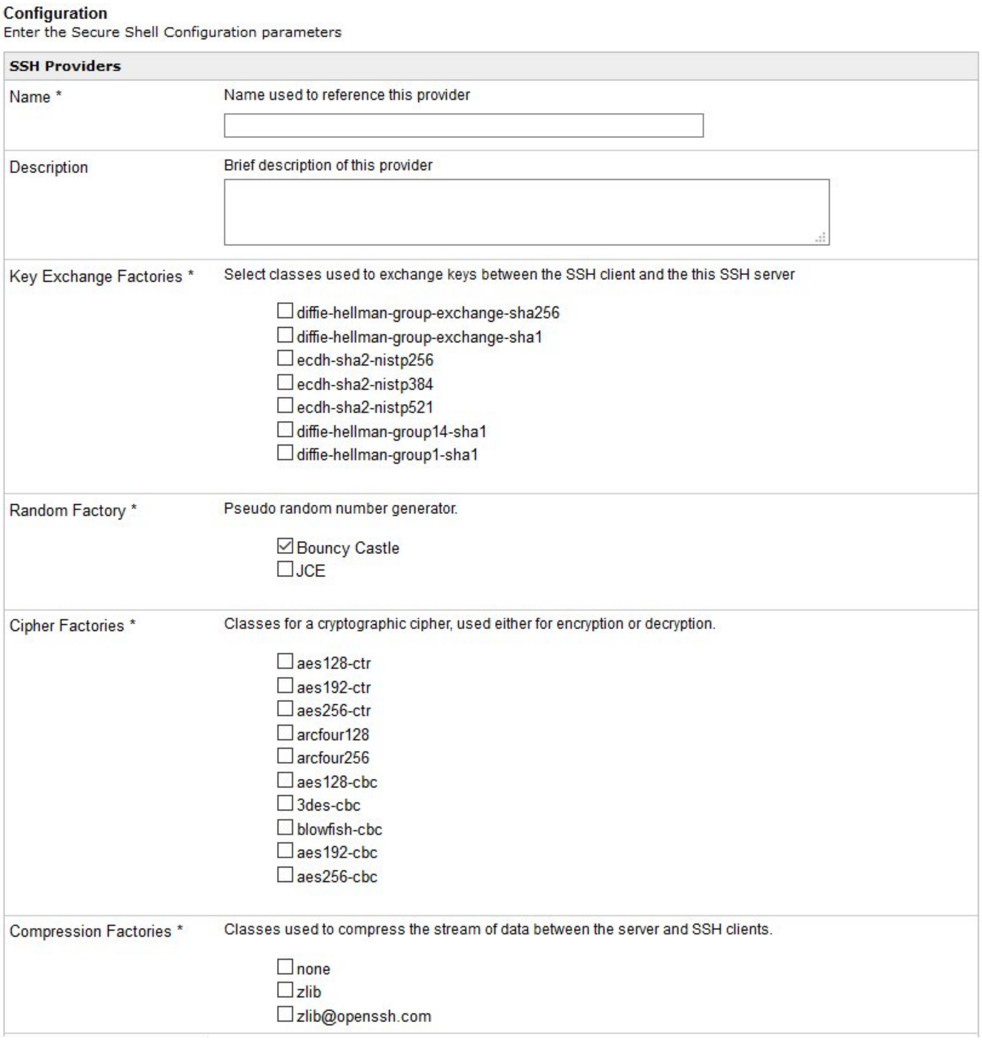
Configuration
Enter the Secure Shell Configuration parameters

Defined SSH Providers
SSH Providers - Provides the parameters needed to configure the SSH Server.

□	Name	Description	Default
□	No SSH Providers have been configured.		

2. Click *New*.

The Configuration pane opens, which contains parameters and settings for the Secure Shell provider, as shown in the following image.



Configuration
Enter the Secure Shell Configuration parameters

SSH Providers	
Name *	Name used to reference this provider
Description	Brief description of this provider
Key Exchange Factories *	Select classes used to exchange keys between the SSH client and the this SSH server □ diffie-hellman-group-exchange-sha256 □ diffie-hellman-group-exchange-sha1 □ ecdh-sha2-nistp256 □ ecdh-sha2-nistp384 □ ecdh-sha2-nistp521 □ diffie-hellman-group14-sha1 □ diffie-hellman-group1-sha1
Random Factory *	Pseudo random number generator. ☑ Bouncy Castle □ JCE
Cipher Factories *	Classes for a cryptographic cipher, used either for encryption or decryption. □ aes128-ctr □ aes192-ctr □ aes256-ctr □ arcfour128 □ arcfour256 □ aes128-cbc □ 3des-cbc □ blowfish-cbc □ aes192-cbc □ aes256-cbc
Compression Factories *	Classes used to compress the stream of data between the server and SSH clients. □ none □ zlib □ zlib@openssh.com

3. Enter the appropriate values for the Secure Shell provider parameters.

4. Click *Add* when you are finished.

   You are returned to the Defined SSH Providers pane, where the new Secure Shell provider that was defined is added to the list.

5. To define multiple Secure Shell providers, repeat this procedure.

Reference: Secure Shell Provider Parameters

The following tables list and describes all of the available parameters and settings for the Secure Shell (SSH) provider.

SSH Providers

Parameter	Description
Name	Name used to reference the Secure Shell (SSH) provider.
Description	Brief description of the Secure Shell (SSH) provider.
Key Exchange Factories	Select one of the following classes that will be used to exchange keys between the SSH client and the SSH server: ❑ diffie-hellman-group-exchange-sha256 ❑ diffie-hellman-group-exchange-sha1 ❑ ecdh-sha2-nistp256 ❑ ecdh-sha2-nistp384 ❑ ecdh-sha2-nistp521 ❑ diffie-hellman-group14-sha1 ❑ diffie-hellman-group1-sha1
Random Factory	Pseudo random number generator: ❑ Bouncy Castle ❑ JCE

Parameter	Description
Cipher Factories	Select one of the following classes for a cryptographic cipher, used either for encryption or decryption: ❑ aes128-ctr ❑ aes192-ctr ❑ aes256-ctr ❑ arcfour128 ❑ arcfour256 ❑ aes128-cbc ❑ 3des-cbc ❑ blowfish-cbc ❑ aes192-cbc ❑ aes256-cbc ❑ none
Compression Factories	Select one of the following classes that will be used to compress the stream of data between the server and SSH clients: ❑ none ❑ zlib ❑ zlib@openssh.com

Parameter	Description
MAC Factories	Select one of the following classes that will be used for Message Authentication Code (MAC) for use in SSH: ❏ hmac-sha2-256 ❏ hmac-sha2-512 ❏ hmac-sha1 ❏ hmac-md5 ❏ hmac-sha1-96 ❏ hmac-md5-96
Signature Factory	Select one of the following classes that will be used by the server to sign and verify packets sent between the server and client. ❏ ecdsa-sha2-nistp256 ❏ ecdsa-sha2-nistp384 ❏ ecdsa-sha2-nistp521 ❏ ssh-dss ❏ ssh-rsa

Key Pair File

Parameter	Description
Key Pair File	

Parameter	Description
Key Pair Provider *	Provider for key pairs. The provider is used to create the SSH Key Pair repository when it doesn't exist. When the repository exists the Provider returns the Key Pair generated by the Signature Factory that was used to create the repository. ❑ com.ibi.sftp.common.keyprovider.XDSecureHostKeyProvider ❑ org.apache.sshd.server.keyprovider.PEMGeneratorHostKeyProvider ❑ org.apache.sshd.server.keyprovider.SimpleGeneratorHostKeyProvider
Key Pair File Signature *	Select one of the following options that will be used by the SFTP Server to sign the Key Pair File that is generated if the Key Pair File does not exist: ❑ RSA ❑ DSA
Key Pair File Path *	Fully qualified path to the Key Pair File. If the path points to a file that does not exist, then the provider will create a Key Pair file at this location using the Key Pair File Signature that was selected.
Key Pair File Password	Password for the SSH Key Pair file.

Authentication

Parameter	Description
Authentication	

Parameter	Description
Password Authenticator *	Select one of the following classes that will be used by the server to authenticate the password of the SSH client: ❑ File Based Password Authenticator ❑ JDBC Based Password Authenticator ❑ RealmBasedPasswordAuthenticator ❑ No Password Authentication ❑ Test Password Authenticator
Public Key Authenticator *	Select one of the following classes that will be used to authenticate the public keys of the SSH client. ❑ Publickey Authenticator ❑ No Public Key Authentication ❑ Test Authenticator

User Repository

Parameter	Description
Authentication Realm	The name of the configured Authentication Realm provider to use. If left blank the user and their authentication credentials must be maintained the SFTPServer User Repository. This field is required if the RealmBasedPasswordAuthenticator is selected.

Parameter	Description
User's Repository Type	Determines how the user repository is stored. The repository can be stored either as an XML file or as a JDBC database. This repository defines the users permitted to exchange messages with this server along with their mailbox and security characteristics. ❑ XML ❑ JDBC ❑ TPM
Repository File	The location of the security file. This field is required either when the Repository Type parameter is set to XML.
Repository Provider	The name of the User Repository's Data Provider. This field is required only when the Repository Type is set to JDBC or TPM.

Basic

Parameter	Description
Basic	
Reuse Address	If set to *true*, when the connection is closed, immediately make the address available, bypassing defaults of TCP.
Connections Backlog *	Number of connections allowed to queue before a failure. The default is 50 connections.

iSM Facilities

In the left console pane of the Server menu, the Facilities group contains links to the following iSM facilities you can configure:

- Activity Facility
- Correlation Facility

The following sections describes how to configure each of the facilities that are available.

Using the Activity Facility

This section provides an overview of the Activity Facility, formerly known as Audit Manager, and describes how to configure it. The iWay Service Manager administrator is responsible for configuring the Activity Facility.

Activity Facility Overview

The Activity Facility now featured in the iWay Service Manager Administration Console was formerly a component of iWay Trading Manager. It maintains a record describing each message that passes through the server. The messages are associated and integrated with the transactions. This makes it possible for an auditor to review them individually or in conjunction with other messages that fall within the scope of the same transaction. Filtering options determine whether the Activity Facility records original input messages, each emitted message, transaction termination status, intermediate activities (for example, parsing and transformations), or all of the above. Activity records enable users to research the status of individual transactions. When used in conjunction with the iWay Enterprise Index (iEI) component or Correlation Facility, the Activity Facility enables these components to retrieve appropriate historic information for their purposes

The following types of Activity Facility handlers are available for configuration:

- **EDI Activity Logs.**
- **iAM 102 Transaction Log Emulator.**
- **iEI Message Manager.** Used for WebFOCUS Magnify integration and for Audit Indexing. Not used for direct indexing. For more information, see *How to Configure the iEI Message Manager Handler* on page 229.
- **Local BAM Driver.**
- **SQL-Based Activity Logs** Preferred Activity Facility handler. For more information, see *How to Configure the SQL-Based Activity Logs Handler* on page 225.
- **Trace logger for test use only.** Used for testing purposes. For more information, see *How to Configure the Trace Log Handler* on page 235.
- **Time event logger for performance measurements.** Used for benchmarking purposes. This handler gathers statistics and produces Microsoft Excel spreadsheets. For more information, see *How to Configure the Time Event Logger* on page 238.

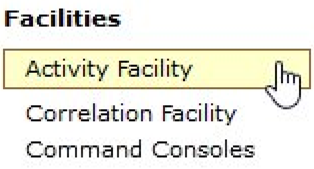
Procedure: How to Configure the SQL-Based Activity Logs Handler

To configure the SQL-Based Activity Logs handler:

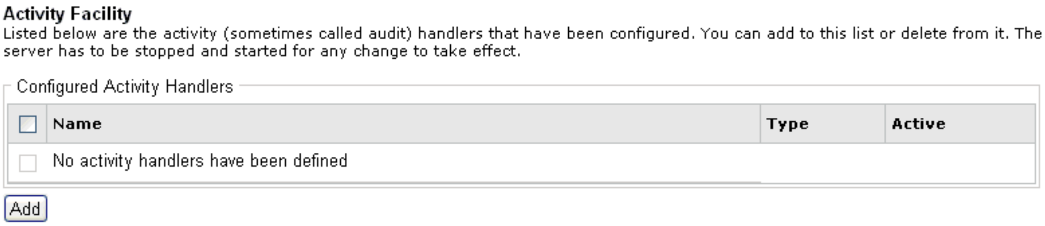
1. Configure a JDBC data provider using the iWay Service Manager Administration Console.

   For more information on configuring data providers, see *Configuring a Data Provider*. Once you have finished, you can continue with configuring the SQL-Based Activity Logs handler.

2. In the left console pane of the Server menu, select *Activity Facility*.



   The Activity Facility pane opens.



Activity Facility
Listed below are the activity (sometimes called audit) handlers that have been configured. You can add to this list or delete from it. The server has to be stopped and started for any change to take effect.

Configured Activity Handlers

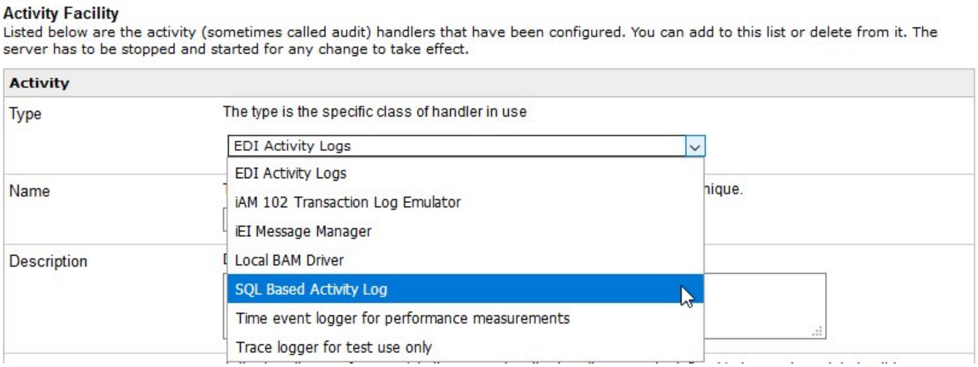
□	Name	Type	Active
□	No activity handlers have been defined		

   The table that is provided lists the configured Activity Facility handlers. Initially, no handlers are shown.

3. Click *Add* to configure a new SQL-Based Activity Logs handler.

   The configuration pane for the Activity Facility handler opens.

4. Select *SQL Based Activity Logs* from the Type drop-down list.

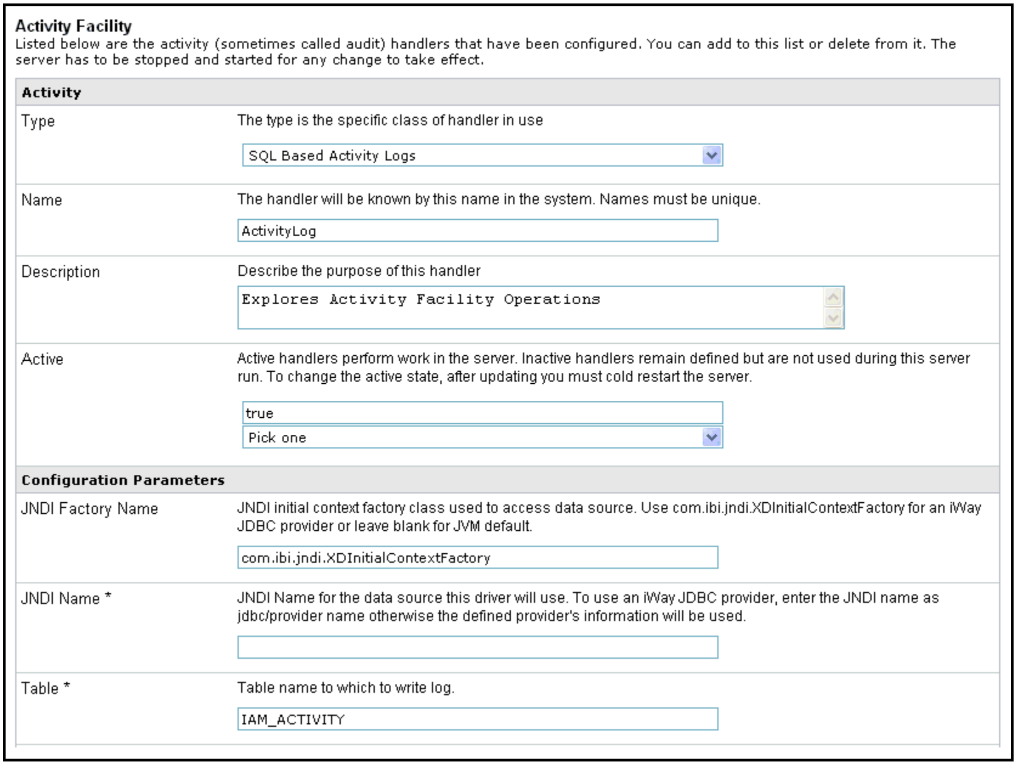


Activity Facility
Listed below are the activity (sometimes called audit) handlers that have been configured. You can add to this list or delete from it. The server has to be stopped and started for any change to take effect.

Activity	
Type	The type is the specific class of handler in use EDI Activity Logs
Name	...nique.
Description	

EDI Activity Logs
iAM 102 Transaction Log Emulator
iEI Message Manager
Local BAM Driver
SQL Based Activity Log
Time event logger for performance measurements
Trace logger for test use only

The Activity Facility pane refreshes and displays the parameters for the SQL-Based Activity Logs handler, as shown in the following image.



Activity Facility

Listed below are the activity (sometimes called audit) handlers that have been configured. You can add to this list or delete from it. The server has to be stopped and started for any change to take effect.

Activity	
Type	The type is the specific class of handler in use SQL Based Activity Logs
Name	The handler will be known by this name in the system. Names must be unique. ActivityLog
Description	Describe the purpose of this handler Explores Activity Facility Operations
Active	Active handlers perform work in the server. Inactive handlers remain defined but are not used during this server run. To change the active state, after updating you must cold restart the server. true Pick one
Configuration Parameters	
JNDI Factory Name	JNDI initial context factory class used to access data source. Use com.ibi.jndi.XDInitialContextFactory for an iWay JDBC provider or leave blank for JVM default. com.ibi.jndi.XDInitialContextFactory
JNDI Name *	JNDI Name for the data source this driver will use. To use an iWay JDBC provider, enter the JNDI name as jdbc/provider name otherwise the defined provider's information will be used.
Table *	Table name to which to write log. IAM_ACTIVITY

5. Provide values for the parameters, as listed and defined in the following table.

Parameter Name	Type	Description
Name (required)	String	Type a unique name for the SQL-Based Activity Logs handler.
Description	String Text Area	Type a description for the SQL-Based Activity Logs handler (optional).

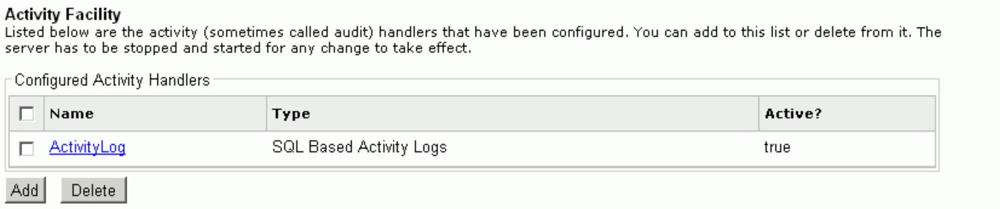
Parameter Name	Type	Description
Active (required)	Boolean Drop-down list	If set to *true*, the SQL-Based Activity Logs handler is activated by default upon server startup. Inactive handlers remain defined but are not automatically activated. You must restart the server to ensure the handler is in an active state.
Configuration Parameters		
JNDI Factory Name	String	JNDI initial context factory class used to access the data source. Use `com.ibi.jndi.XDInitialContextFactory` for an iWay JDBC provider or leave blank for JVM default.
JNDI Name (required)	String	JNDI name for the data source that is used by the driver. To use an iWay JDBC provider, enter the JNDI name as `jdbc/provider_name` otherwise the defined information for the provider will be used.
Table (required)	String	Table name for the activity log. This must be a valid identifier in the database being used. If the table does not exist at startup, it will be created automatically.
Compression	Drop-down list	Specify whether the messages are to be compressed. Values include: ❏ none (default) ❏ smallest ❏ fastest ❏ standard ❏ Huffman

Parameter Name	Type	Description
Start Events (required)	Boolean Drop-down list	If set to *true* (default), the input messages will be recorded in the activity log. This value must be set to *true* for use of the audit reports in the console.
Internal Events (required)	Boolean Drop-down list	If set to *true*, system events are included in the activity log. System events include activities, such as parsing and transformations (optional). False is selected by default.
Security Events (required)	Boolean Drop-down list	If set to *true* (default), security events are recorded. This includes digital signature and so on. This does not include console activity.
Business Error Events (required)	Boolean Drop-down list	If set to *true*, business errors are recorded, such as rules system violations. False is selected by default.
Emit Events (required)	Boolean Drop-down list	If set to *true* (default), output messages from emitter services will be recorded. This is required for use of the audit log reports in the console.
End Events (required)	Boolean Drop-down list	If set to *true* (default), the end of message processing will be recorded in the activity log. This is required for use of the audit log reports in the console.
Notes Table (required)	String	Table name for the notes table, which contains log annotations. If the table does not exist at startup, it will be created automatically.
MAC Algorithm	String Drop-down list	The Message Authentication Code (MAC) algorithm. None (default) indicates a MAC should not be computed.
MAC Provider	String Drop-down list	The Message Authentication Code (MAC) provider. Not Specified indicates the default provider should be used. The remaining available value is *SunJCE*.

Parameter Name	Type	Description
MAC Secret Key	String	The Message Authentication Code (MAC) secret key to use.

6. Click *Update*.

   You are returned to main Activity Facility pane where the newly configured SQL-Based Activity Logs handler is added to the list, as shown in the following image.



Activity Facility

Listed below are the activity (sometimes called audit) handlers that have been configured. You can add to this list or delete from it. The server has to be stopped and started for any change to take effect.

Configured Activity Handlers

□	Name	Type	Active?
□	ActivityLog	SQL Based Activity Logs	true

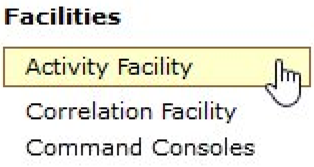
Procedure: How to Configure the iEI Message Manager Handler

To configure the iEI Message Manager handler:

1. Configure a JDBC data provider using the iWay Service Manager Administration Console.

   For more information on configuring data providers, see *Configuring a Data Provider*. Once you have finished, you can continue with configuring the iEI Message Manager handler.

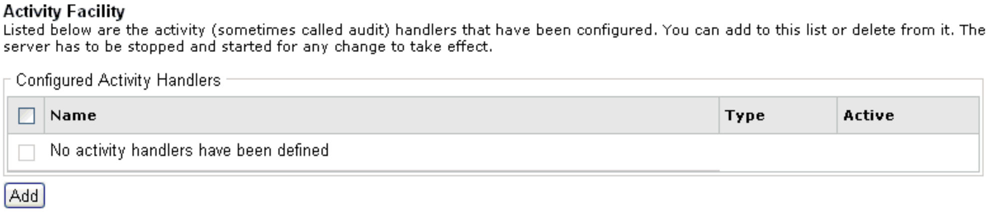
2. In the left console pane of the Server menu, select *Activity Facility*.



   The Activity Facility pane opens.

   The table that is provided lists the configured Activity Facility handlers. Initially, no handlers are shown.

3. Click *Add* to configure a new iEI Message Manager handler.



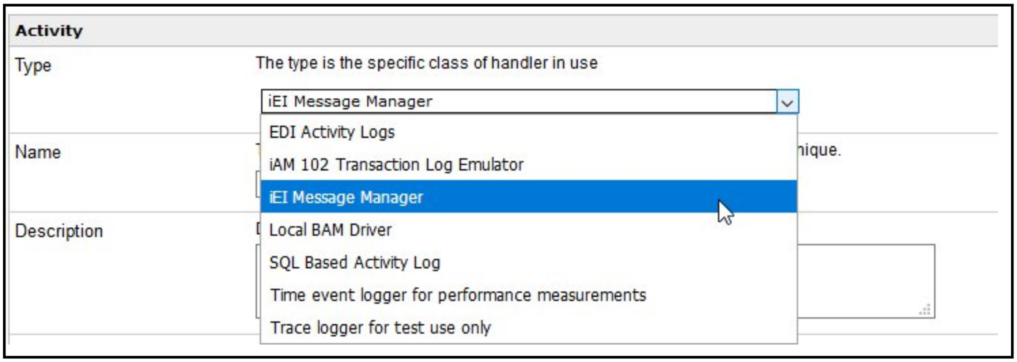
Activity Facility
Listed below are the activity (sometimes called audit) handlers that have been configured. You can add to this list or delete from it. The server has to be stopped and started for any change to take effect.

Configured Activity Handlers

□	Name	Type	Active
□	No activity handlers have been defined		

The configuration pane for the Activity Facility handler opens.

4. Select *iEI Message Manager* from the Type drop-down list.



Activity

Type: The type is the specific class of handler in use

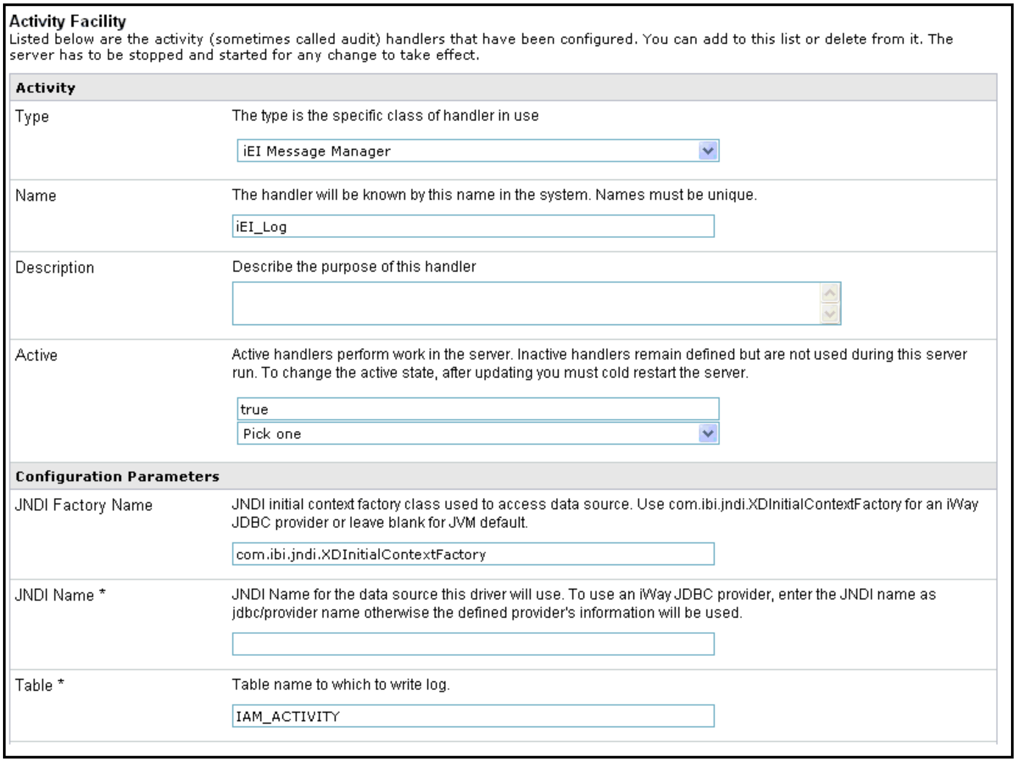
iEI Message Manager

Name

Description

EDI Activity Logs
iAM 102 Transaction Log Emulator
iEI Message Manager
Local BAM Driver
SQL Based Activity Log
Time event logger for performance measurements
Trace logger for test use only
nique.

The Activity Facility pane refreshes and displays the parameters for the iEI Message Manager handler, as shown in the following image.



Activity Facility

Listed below are the activity (sometimes called audit) handlers that have been configured. You can add to this list or delete from it. The server has to be stopped and started for any change to take effect.

Activity	
Type	The type is the specific class of handler in use iEI Message Manager
Name	The handler will be known by this name in the system. Names must be unique. iEI_Log
Description	Describe the purpose of this handler
Active	Active handlers perform work in the server. Inactive handlers remain defined but are not used during this server run. To change the active state, after updating you must cold restart the server. true Pick one
Configuration Parameters	
JNDI Factory Name	JNDI initial context factory class used to access data source. Use com.ibi.jndi.XDInitialContextFactory for an iWay JDBC provider or leave blank for JVM default. com.ibi.jndi.XDInitialContextFactory
JNDI Name *	JNDI Name for the data source this driver will use. To use an iWay JDBC provider, enter the JNDI name as jdbc/provider name otherwise the defined provider's information will be used.
Table *	Table name to which to write log. IAM_ACTIVITY

5. Provide values for the parameters, as listed and defined in the following table.

Parameter Name	Type	Description
Name (required)	String	Type a unique name for the iEI Message Manager handler.
Description	String Text Area	Type a description for the iEI Message Manager handler (optional).

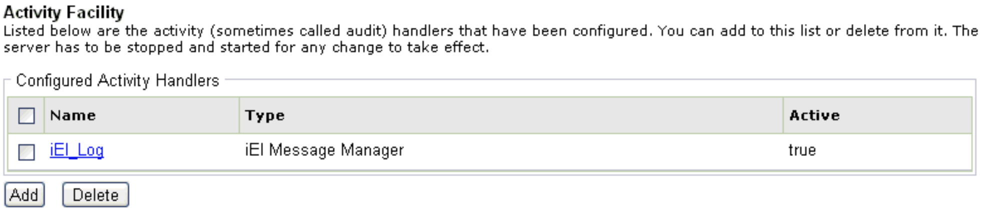
Parameter Name	Type	Description
Active (required)	Boolean Drop-down list	If set to *true*, the iEI Message Manager handler is activated by default upon server startup. Inactive handlers remain defined but are not automatically activated. You must restart the server to ensure the handler is in an active state.
Configuration Parameters		
JNDI Factory Name	String	JNDI initial context factory class used to access the data source. Use `com.ibi.jndi.XDInitialContextFactory` for an iWay JDBC provider or leave blank for JVM default.
JNDI Name (required)	String	JNDI name for the data source that is used by the driver. To use an iWay JDBC provider, enter the JNDI name as `jdbc/provider_name` otherwise the defined information for the provider will be used.
Table (required)	String	Table name for the activity log. This must be a valid identifier in the database being used. If the table does not exist at startup, it will be created automatically.
Compression	Drop-down list	Specify whether the messages are to be compressed. Values include: ❑ none (default) ❑ smallest ❑ fastest ❑ standard ❑ Huffman

Parameter Name	Type	Description
Start Events (required)	Boolean Drop-down list	If set to *true* (default), the input messages will be recorded in the activity log. This value must be set to *true* for use of the audit reports in the console.
Internal Events (required)	Boolean Drop-down list	If set to *true*, system events are included in the activity log. System events include activities, such as parsing and transformations (optional). False is selected by default.
Business Error Events (required)	Boolean Drop-down list	If set to *true*, business errors are recorded, such as rules system violations. False is selected by default.
Emit Events (required)	Boolean Drop-down list	If set to *true* (default), output messages from emitter services will be recorded. This is required for use of the audit log reports in the console.
End Events (required)	Boolean Drop-down list	If set to *true* (default), the end of message processing will be recorded in the activity log. This is required for use of the audit log reports in the console.
Notes Table (required)	String	Table name for the notes table, which contains log annotations. If the table does not exist at startup, it will be created automatically.
iEI		
URL for the search appliance (required)	String	URL for the search appliance.
Feed Datasource (required)	String	Data source for search appliance feeds.

Parameter Name	Type	Description
Base URL (required)	String	Base URL for indexed feeds. The default base URL is: `http://*:9996/mm`
URL Add-on	String	Additional ampersand-delimited key=value pairs to append to the base URL for indexed feeds. SREG() is evaluated
Batch Size (required)	String	Number of records to add to the feed before submitting to search appliance. The default value is 2.
Secure Search? (required)	Boolean Drop-down list	If set to *true* (default), HTTP basic authentication is used to secure the feed and queries.
Index Start Events (required)	Boolean Drop-down list	If set to *true* (default), start event is indexed.
Index Emit Events (required)	Boolean Drop-down list	If set to *true* (default), emit event is indexed.

6. Click *Update*.

   You are returned to main Activity Facility pane where the newly configured iEI Message Manager handler is added to the list, as shown in the following image.



Activity Facility

Listed below are the activity (sometimes called audit) handlers that have been configured. You can add to this list or delete from it. The server has to be stopped and started for any change to take effect.

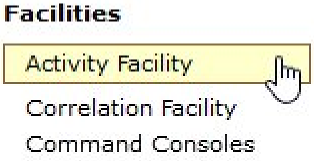
Configured Activity Handlers

□	Name	Type	Active
□	iEI_Log	iEI Message Manager	true

Procedure: How to Configure the Trace Log Handler

To configure the Trace Log handler:

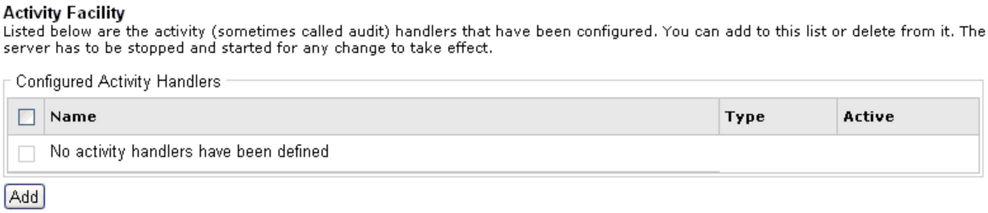
1. In the left console pane of the Server menu, select *Activity Facility*.



The Activity Facility pane opens.

The table that is provided lists the configured Activity Facility handlers. Initially, no handlers are shown.

2. Click *Add* to configure a new Trace Log handler.



Activity Facility

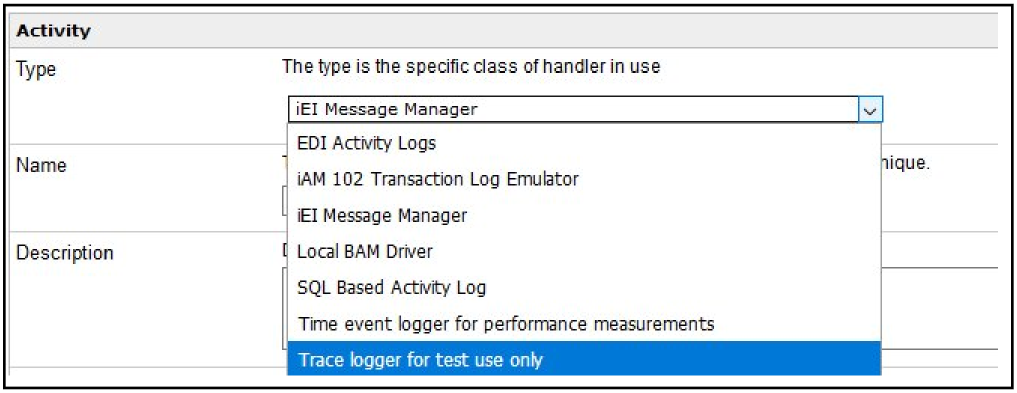
Listed below are the activity (sometimes called audit) handlers that have been configured. You can add to this list or delete from it. The server has to be stopped and started for any change to take effect.

Configured Activity Handlers

□	Name	Type	Active
□	No activity handlers have been defined		

The configuration pane for the Activity Facility handler opens.

3. Select *Trace logger for test use only* from the Type drop-down list.



Activity

Type — The type is the specific class of handler in use

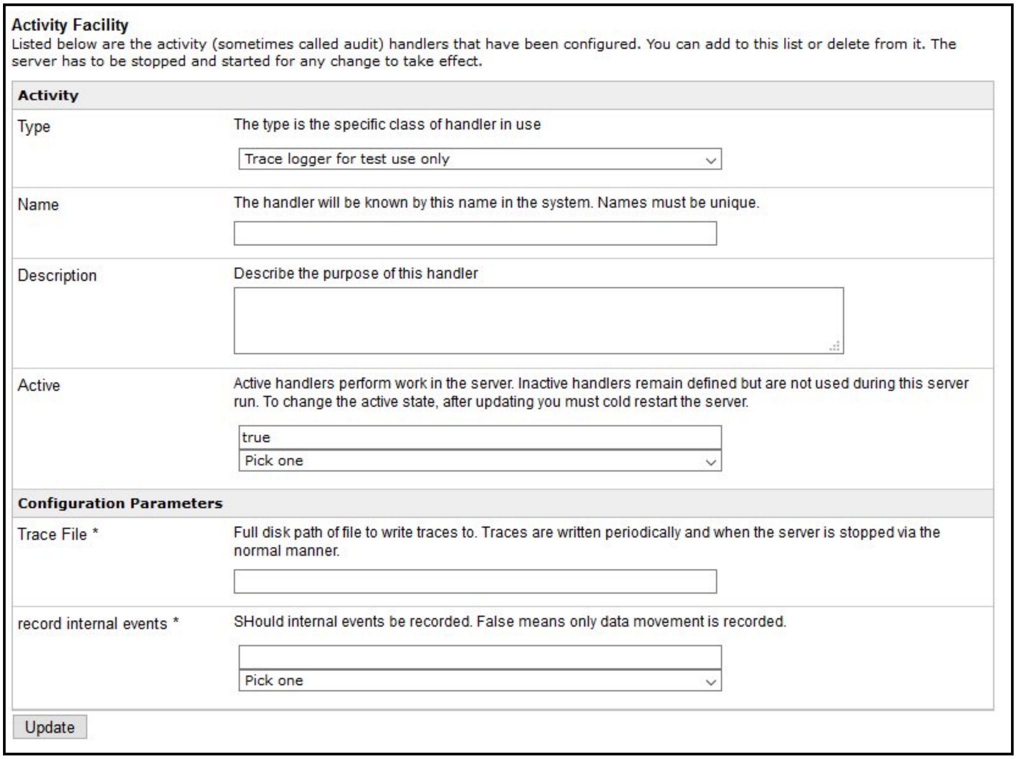
iEI Message Manager

Name — …nique.

Description

EDI Activity Logs
iAM 102 Transaction Log Emulator
iEI Message Manager
Local BAM Driver
SQL Based Activity Log
Time event logger for performance measurements
Trace logger for test use only

The Activity Facility pane refreshes and displays the parameters for the Trace Log handler, as shown in the following image.



Activity Facility
Listed below are the activity (sometimes called audit) handlers that have been configured. You can add to this list or delete from it. The server has to be stopped and started for any change to take effect.

Activity	
Type	The type is the specific class of handler in use Trace logger for test use only
Name	The handler will be known by this name in the system. Names must be unique.
Description	Describe the purpose of this handler
Active	Active handlers perform work in the server. Inactive handlers remain defined but are not used during this server run. To change the active state, after updating you must cold restart the server. true Pick one
Configuration Parameters	
Trace File *	Full disk path of file to write traces to. Traces are written periodically and when the server is stopped via the normal manner.
record internal events *	SHould internal events be recorded. False means only data movement is recorded. Pick one

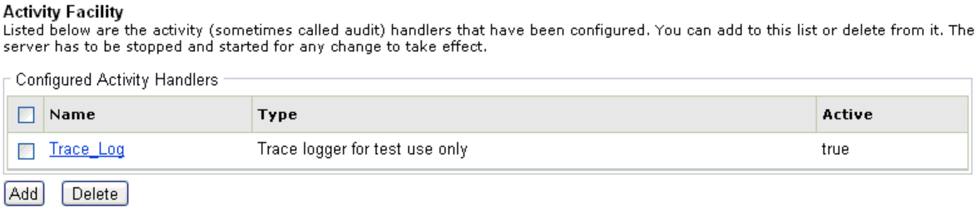
4. Provide values for the parameters, as listed and defined in the following table.

Parameter Name	Type	Description
Name (required)	String	Type a unique name for the Trace Log handler.
Description	String Text Area	Type a description for the Trace Log handler (optional).

Parameter Name	Type	Description
Active (required)	Boolean Drop-down list	If set to *true*, the Trace Log handler is activated by default upon server startup. Inactive handlers remain defined but are not automatically activated. You must restart the server to ensure the handler is in an active state.
Configuration Parameters		
Trace File	String	Path to the file where the trace log will be written.
record internal events *	Boolean	Determines whether internal events should be recorded. False means only data movement is recorded.

5. Click *Update*.

   You are returned to main Activity Facility pane where the newly configured Trace Log handler is added to the list, as shown in the following image.



Activity Facility

Listed below are the activity (sometimes called audit) handlers that have been configured. You can add to this list or delete from it. The server has to be stopped and started for any change to take effect.

Configured Activity Handlers

□	Name	Type	Active
□	Trace_Log	Trace logger for test use only	true

Using the Time Event Logger (XDTimeLogger)

A special activity log exit, called the Time Event Logger, is now included among the standard activity log exits. This logger can be used in performance measuring situations, but is not intended itself to be a full performance measuring solution. The Time Event Logger logs statistics for each message and internal event executed in the iWay Service Manager (iSM) server. In iSM, an event is the execution of a component, such as a transform, rule validation, process flow, or service within the flow.

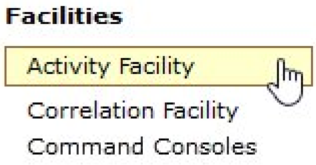
Log exits can also record other events, such as error messages, business errors detected by a process, an emit of a message, and so on. The Time Event Logger does not record this information. Similarly, this exit is unconcerned about the status of the message during its flow. You are expected to have structured the performance test such that useful information can be extracted from the measurements that are recorded.

The Time Event Logger generates an output log file intended for use in a standard spreadsheet application. There is no restriction on how the data is actually analyzed.

Procedure: How to Configure the Time Event Logger

To configure the Time Event Logger:

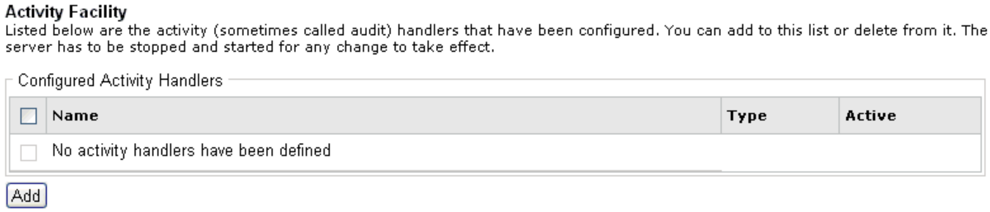
1. In the left console pane of the Server menu, select *Activity Facility*.



The Activity Facility pane opens.

The table that is provided lists the configured Activity Facility handlers. Initially, no handlers are shown.

2. Click *Add*.



Activity Facility

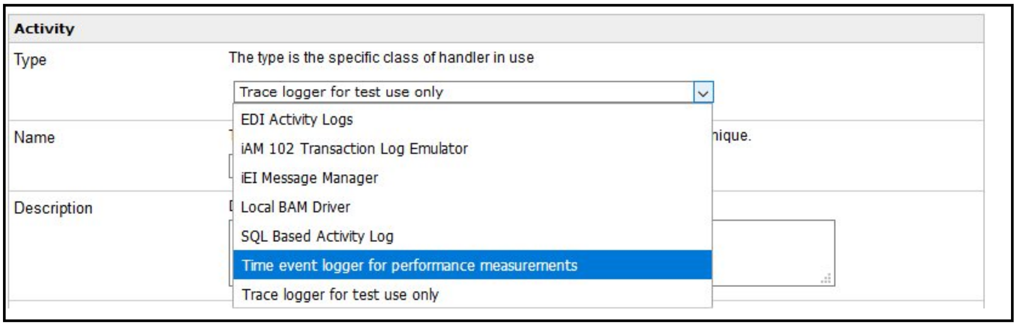
Listed below are the activity (sometimes called audit) handlers that have been configured. You can add to this list or delete from it. The server has to be stopped and started for any change to take effect.

Configured Activity Handlers

□	Name	Type	Active
□	No activity handlers have been defined		

The configuration pane for the Activity Facility handler opens.

3. From the Type drop-down list, select *Time event logger for performance measurements*.



Activity

Type

The type is the specific class of handler in use

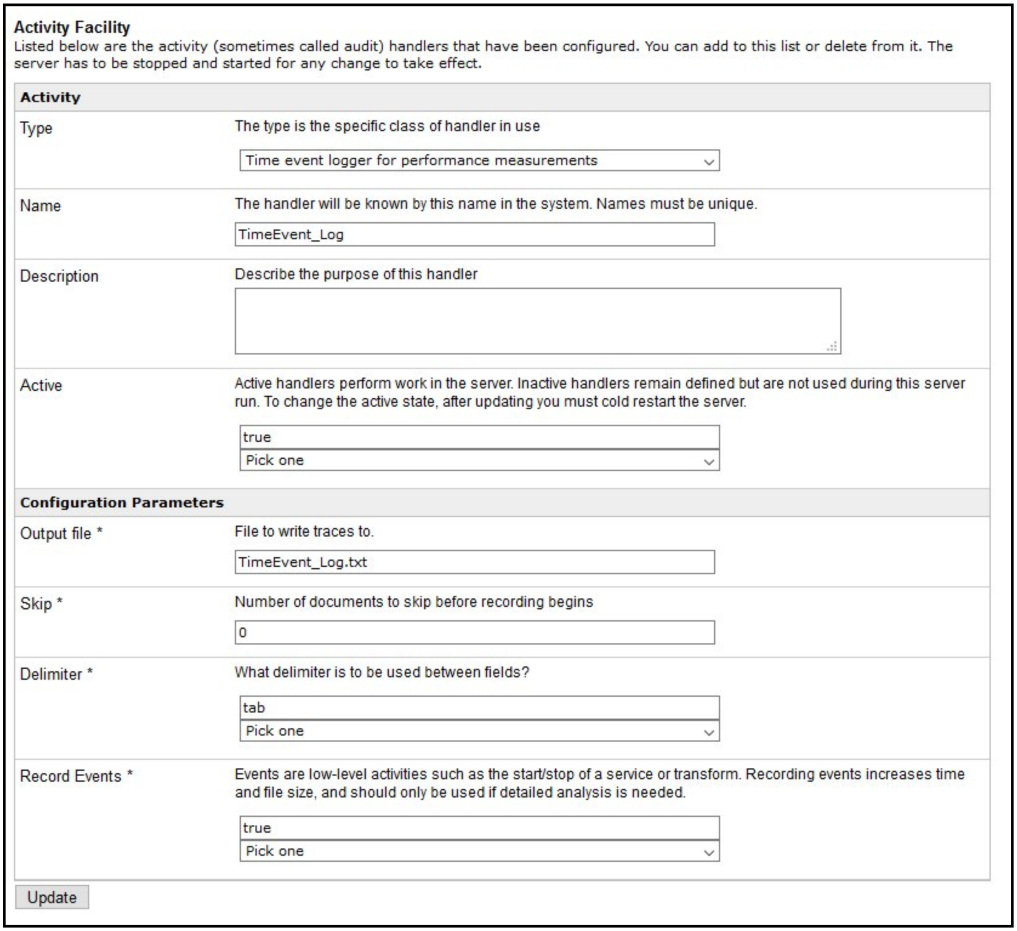
Trace logger for test use only

Name

Description

EDI Activity Logs
iAM 102 Transaction Log Emulator
iEI Message Manager
Local BAM Driver
SQL Based Activity Log
Time event logger for performance measurements
Trace logger for test use only
nique.

The Activity Facility pane refreshes and displays the configuration parameters for the Time Event Logger, as shown in the following image.



Activity Facility

Listed below are the activity (sometimes called audit) handlers that have been configured. You can add to this list or delete from it. The server has to be stopped and started for any change to take effect.

Activity

Type

The type is the specific class of handler in use

Time event logger for performance measurements

Name

The handler will be known by this name in the system. Names must be unique.

TimeEvent_Log

Description

Describe the purpose of this handler

Active

Active handlers perform work in the server. Inactive handlers remain defined but are not used during this server run. To change the active state, after updating you must cold restart the server.

true

Pick one

Configuration Parameters

Output file *

File to write traces to.

TimeEvent_Log.txt

Skip *

Number of documents to skip before recording begins

0

Delimiter *

What delimiter is to be used between fields?

tab

Pick one

Record Events *

Events are low-level activities such as the start/stop of a service or transform. Recording events increases time and file size, and should only be used if detailed analysis is needed.

true

Pick one

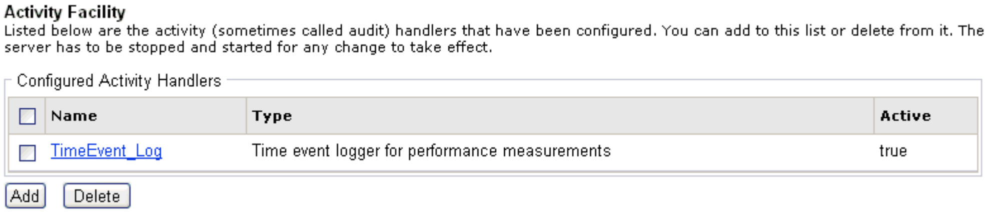
4. Provide values for the parameters, which are listed and defined in the following table.

Parameter Name	**Type**	**Description**
Name (required)	String	Type a unique name for the Time Event Logger.
Description	String Text Area	Type a description for the Time Event Logger (optional).
Active (required)	Boolean Drop-down list	If set to *true*, the Time Event Logger is activated by default upon server startup. Inactive handlers remain defined but are not automatically activated. You must restart the server to ensure the handler is in an active state.
Configuration Parameters		
Output file	String	This is the file where traces are written. As the server starts, a new instance of this file is created. It is the responsibility of the user to save the log file for any server sessions that require further examination.

Parameter Name	Type	Description
Skip	String	This is the number of transactions to skip before actual recording starts. When performing runs for relative measures of performance, it is recommended that at least a few transactions not be recorded. This is because the server often performs "lazy initialization" during the first few transactions, so that these are slower than subsequent transactions as routes are established, services are loaded, process flows compiled, and so on. When testing for measures of more absolute performance, it is important that many transactions be skipped. The Java HotSpot VM uses the first transactions to gather statistics before optimization of the application. In the HotSpot documentation for Java 5.0, Sun recommends 10000 transactions.
Delimiter	Boolean Drop-down list	Fields in the output file are separated by a delimiter character. You can select either tab or comma. Different spreadsheet and data analysis programs may prefer one or the other. The default delimiter is tab.
Record Events *	Boolean	Events are low-level activities, such as the start/stop of a service or a transform. Recording events increases time and file size, and should only be used if detailed analysis is needed.

5. Click *Update*.

You are returned to main Activity Facility pane where the newly configured Time Event Logger is added to the list, as shown in the following image.



Activity Facility

Listed below are the activity (sometimes called audit) handlers that have been configured. You can add to this list or delete from it. The server has to be stopped and started for any change to take effect.

Configured Activity Handlers

□	Name	Type	Active
□	TimeEvent_Log	Time event logger for performance measurements	true

Recorded Information

The Time Event Logger records information pertaining to channel and event execution. As the exit is initialized, a header (often called a DIF header) is written to the output. This represents the column heads for the data to be recorded.

The following table lists and describes the data that is recorded.

Recorded Data	Description
Type	This is the type of event being recorded (either *chan* or *evnt*). A channel record (*chan*) is the passage of a message from receipt to completion. An event record (*evnt*) is one activity that takes place during the processing of the message.
Key	This is a relatively unique key, used to identify the record in the logger. It is used to relate the start of the event to the end of the event, and is not guaranteed to be unique for the duration of the run. It is most likely of little use during data analysis.
Event	The event identifier. This is an integer used in the iSM server to identify specific events.
Duration	The wall clock time from the start of the event to its end.
CPU	The CPU time associated with the event. For this time to be available, the iSM server must be able to record detailed statistics. CPU time is the total amount of time the JVM spends in the thread of execution of the operation.

Recorded Data	Description
User	The user time associated with the event. For this time to be available, the iSM server must be able to record detailed statistics. The user time is the amount of time spent in application (server) code, not including JVM execution time.
Name	The name of the event. This is used to document the record.
Extra	Some events may include extra information. For example, an event recording the execution of a service agent will include the name of the agent.
Threaded	The identifier of the execution thread.
Initiate Time GMT	The start time of the event, in GMT.
Initiate Time ms	The Unix Epoch time, extended to milliseconds, at which the event starts.

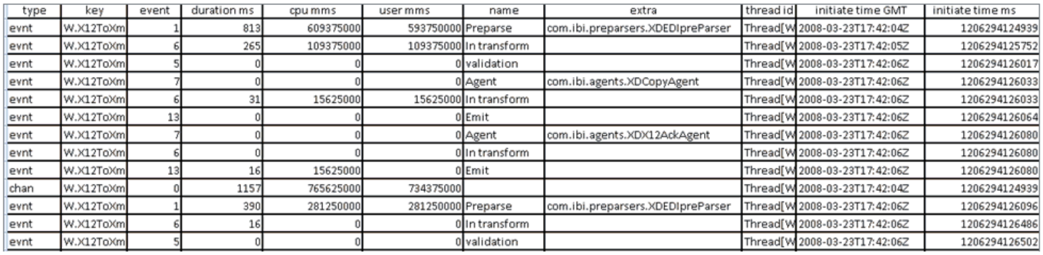
An iSM server is capable of recording extended statistics only if the iwmeasure.jar file is installed in the following directory:

```
<iwayhome>/etc/manager/extensions
```

Note: The iwmeasure.jar file is only available by request from iWay Software.

Analyzing Generated Data

The generated data can be loaded into a spreadsheet program for data analysis. A sample is shown in the following image.



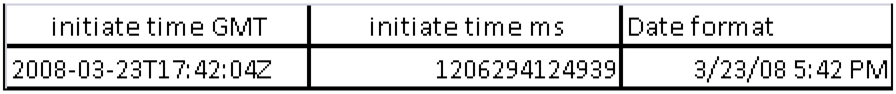
type	key	event	duration ms	cpu mms	user mms	name	extra	thread id	initiate time GMT	initiate time ms
evnt	W.X12ToXm	1	813	609375000	593750000	Preparse	com.ibi.preparsers.XDEDIpreParser	Thread[W	2008-03-23T17:42:04Z	1206294124939
evnt	W.X12ToXm	6	265	109375000	109375000	In transform		Thread[W	2008-03-23T17:42:05Z	1206294125752
evnt	W.X12ToXm	5	0	0	0	validation		Thread[W	2008-03-23T17:42:06Z	1206294126017
evnt	W.X12ToXm	7	0	0	0	Agent	com.ibi.agents.XDCopyAgent	Thread[W	2008-03-23T17:42:06Z	1206294126033
evnt	W.X12ToXm	6	31	15625000	15625000	In transform		Thread[W	2008-03-23T17:42:06Z	1206294126033
evnt	W.X12ToXm	13	0	0	0	Emit		Thread[W	2008-03-23T17:42:06Z	1206294126064
evnt	W.X12ToXm	7	0	0	0	Agent	com.ibi.agents.XDX12AckAgent	Thread[W	2008-03-23T17:42:06Z	1206294126080
evnt	W.X12ToXm	6	0	0	0	In transform		Thread[W	2008-03-23T17:42:06Z	1206294126080
evnt	W.X12ToXm	13	16	15625000	0	Emit		Thread[W	2008-03-23T17:42:06Z	1206294126080
chan	W.X12ToXm	0	1157	765625000	734375000			Thread[W	2008-03-23T17:42:04Z	1206294124939
evnt	W.X12ToXm	1	390	281250000	281250000	Preparse	com.ibi.preparsers.XDEDIpreParser	Thread[W	2008-03-23T17:42:06Z	1206294126096
evnt	W.X12ToXm	6	16	0	0	In transform		Thread[W	2008-03-23T17:42:06Z	1206294126486
evnt	W.X12ToXm	5	0	0	0	validation		Thread[W	2008-03-23T17:42:06Z	1206294126502

Notice that information is recorded upon completion of the event that is being timed. So you can see the channel record (*chan*) following the events within that channel.

The duration column is recorded in milliseconds, while the cpu and user columns are recorded in microseconds. The actual precision of the recording depends upon the timing characteristics of the processor on which the JVM is running. If the iSM server is not running with the iwmeasure statistics package, cpu and user columns will always be zero. The timings in this example frequently show zeros, however, because the resolution of the clocks shows the difference between the start and end time for the event to be too fast to measure.

The **initiate time ms** column records the start of event time in Unix Epoch milliseconds. This can be converted to a standard Excel time column by using the following formula:

```
=(<cell>/86400000)+25569)
```



initiate time GMT	initiate time ms	Date format
2008-03-23T17:42:04Z	1206294124939	3/23/08 5:42 PM

This formula accounts for the conversion from the Epoch time to Excel time by determining the number of seconds in the period and adjusting for the difference in the Epoch and Excel starting points (1900 vs. 1970). This example shows how the application of this formula makes the two columns (initiate time GMT and Date format) produce the same values. Naturally, the column can be used for other types of data analysis. This is only an example.

Using the Correlation Facility

The Correlation Facility tracks related events over time. It associates messages into groups for business purposes and provides services to:

- Associate incoming asynchronous responses with the appropriate outgoing message. An example is an asynchronous MDN for an AS2 message.
- Group incoming messages together for business purposes. An example is an HL7 Storyboard associating related health messages and their responses. Such groups have different names in various protocols, and are called Correlation Sets in iWay terminology.

Note: Do not configure the Correlation Facility if you are using iWay Business Activity Monitor (BAM). iWay BAM automatically configures this functionality.

Messages

Messages that are managed by the Correlation Facility pass through a variety of states depending upon the purpose and protocol associated with the message. These states include:

- Open
- Technical ACK or NAK

- Business ACK or NAK
- Anticipating response
- Closed

Associated with each stored message is the transaction identifier, which links the message to the audit facility, along with state, timestamps, and user information that can be recovered and used to control a subsequent process flow.

Services

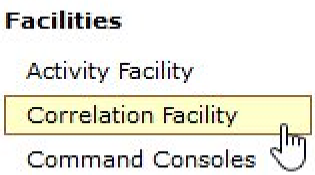
The Correlation Facility provides a variety of services available during a process flow:

- Open a Correlation or Correlation Set and associate the current message with that set.
- Update a message's state.
- Add a message to a correlation set.
- Close a correlation set.
- Locate correlations and correlation sets in specific states.
- Retrieve information for a correlation or correlation set, including:
    - State
    - Transaction (message) information
    - User information (application-specific)

Procedure: How to Configure the Correlation Facility

To configure the Correlation Facility:

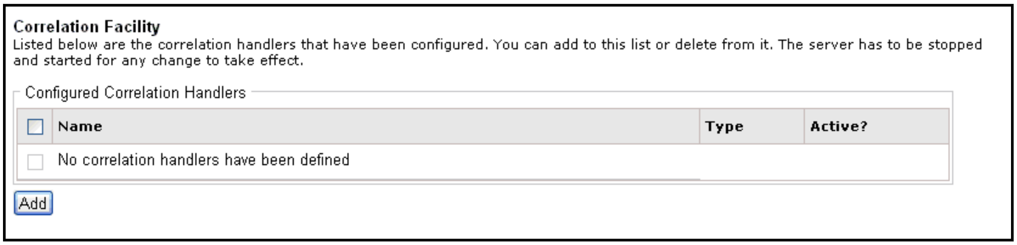
1. In the left console pane of the Server menu, select *Correlation Facility*.

   The Correlation Facility pane opens.

   The table that is provided lists the configured Correlation Facility handlers. Initially, no handlers are shown.

2. Click *Add* to configure a new Correlation Facility handler.



Correlation Facility

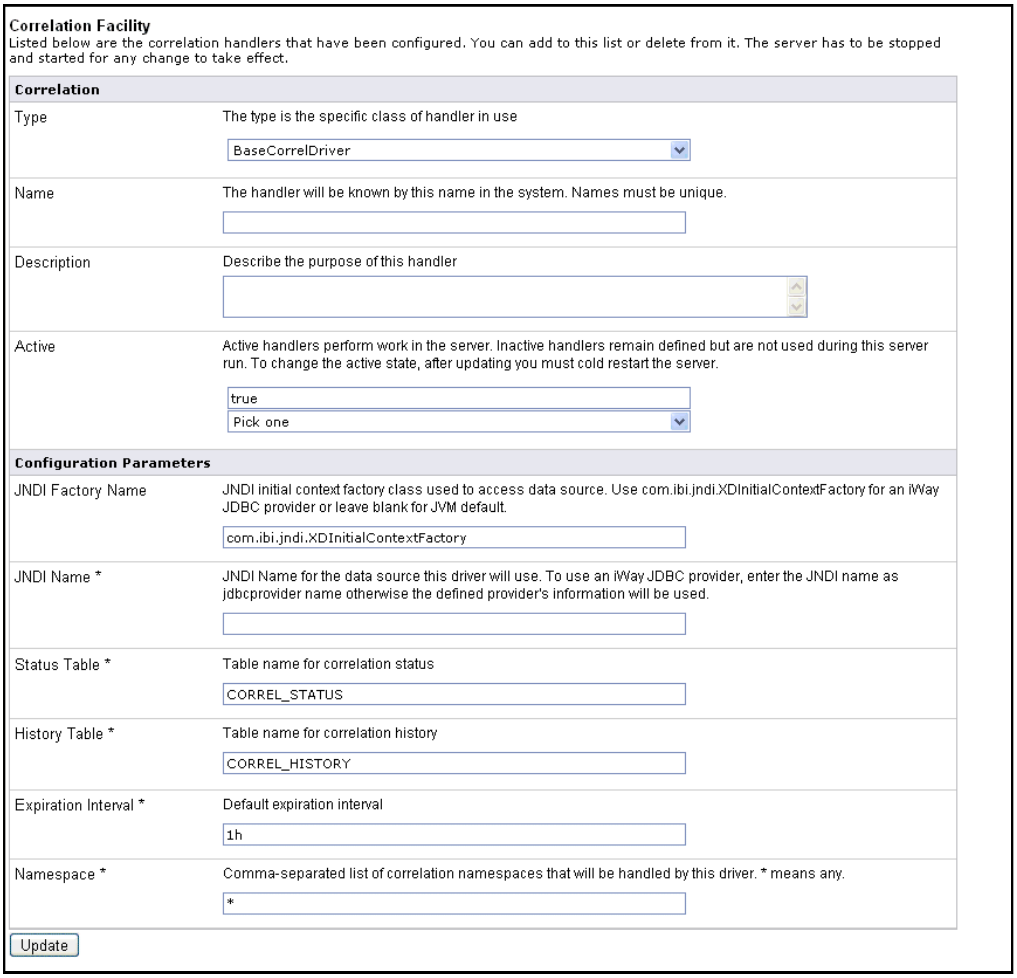
Listed below are the correlation handlers that have been configured. You can add to this list or delete from it. The server has to be stopped and started for any change to take effect.

Configured Correlation Handlers

□	Name	Type	Active?
□	No correlation handlers have been defined		

The configuration pane for the Correlation Facility handler opens.

3. From the Type drop-down list, select a type of handler. The default value is *BaseCorrelDriver*.



Correlation Facility

Listed below are the correlation handlers that have been configured. You can add to this list or delete from it. The server has to be stopped and started for any change to take effect.

Correlation		
Type	The type is the specific class of handler in use	BaseCorrelDriver
Name	The handler will be known by this name in the system. Names must be unique.	
Description	Describe the purpose of this handler	
Active	Active handlers perform work in the server. Inactive handlers remain defined but are not used during this server run. To change the active state, after updating you must cold restart the server.	true Pick one
Configuration Parameters		
JNDI Factory Name	JNDI initial context factory class used to access data source. Use com.ibi.jndi.XDInitialContextFactory for an iWay JDBC provider or leave blank for JVM default.	com.ibi.jndi.XDInitialContextFactory
JNDI Name *	JNDI Name for the data source this driver will use. To use an iWay JDBC provider, enter the JNDI name as jdbcprovider name otherwise the defined provider's information will be used.	
Status Table *	Table name for correlation status	CORREL_STATUS
History Table *	Table name for correlation history	CORREL_HISTORY
Expiration Interval *	Default expiration interval	1h
Namespace *	Comma-separated list of correlation namespaces that will be handled by this driver. * means any.	*

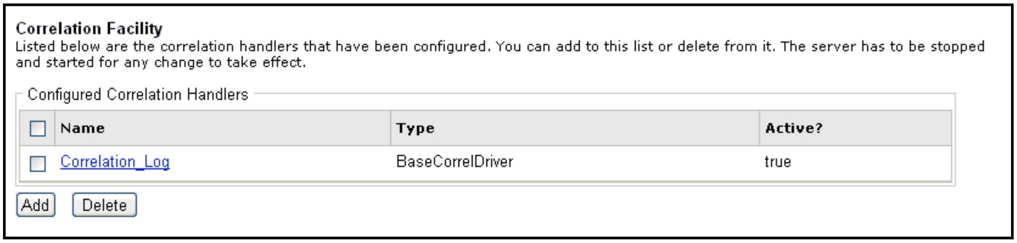
4. Provide values for the parameters, as defined in the following table.

Parameter Name	Type	Description
Correlation		
Type	String Drop-down list	Type of Correlation Facility handler to use.
Name	String	Unique name for the selected handler.
Description	String Text Area	Description for this handler (optional).
Active	Boolean Drop-down list	If set to *true*, the handler is activated by default upon server startup. Inactive handlers remain defined but are not automatically activated. You must restart the server to ensure the handler is in an active state.
Configuration Parameters		
JNDI Factory Name	String	JNDI initial context factory class used to access the data source. Use `com.ibi.jndi.XDInitialContextFactory` for an iWay JDBC provider or leave blank for JVM default.
JNDI Name (required)	String	JNDI name for the data source that is used by the driver. To use an iWay JDBC provider, enter the JNDI name as `jdbc/provider_name` otherwise the defined information for the provider will be used.
Status Table (required)	String	Table name for correlation status. The default value is CORREL_STATUS.

Parameter Name	Type	Description
History Table (required)	String	Table name for the correlation history. The default value is CORREL_HISTORY.
Expiration Interval (required)	String	Default expiration interval. One hour (1h) is the default value.
Namespace (required)	String	Comma-separated list of correlation namespaces that will be handled by this driver. An asterisk character (*) accepts any namespace (default).

5. Click *Update*.

   You are returned to the main Correlation Facility pane where the newly configured handler is added to the list, as shown in the following image.



Correlation Facility

Listed below are the correlation handlers that have been configured. You can add to this list or delete from it. The server has to be stopped and started for any change to take effect.

Configured Correlation Handlers

□	Name	Type	Active?
□	Correlation_Log	BaseCorrelDriver	true

Chapter 7

Configuring iWay Registry Components

The following topics describe how to configure iWay business components using the iWay Service Manager Administration Console. Each iWay business component is stored in the registry, which supports the design-time activities of the server.

In this chapter:

Understanding the iWay Registry

The iWay Service Manager (iSM) Administration Console provides a central interface for design-time and run-time activities, which are independent from one another. During design time, components can be constructed and configured without referencing any specific run-time servers. These components are stored in a design-time registry. Once design-time components have been defined, they can be assembled and deployed to one or more run-time instances of iSM. The registry also gives access to the configuration of resources such as adapters, emitters, listeners, services, and encryptors. Please note that there is only one registry that is available for each instance of iWay Service Manager.

Adding an Adapter

Before using an adapter in iWay Service Manager (iSM), you must create a target for it, which represents a specific instance of a connection to a back-end system. For more information on creating targets and connections using iWay Explorer, see the corresponding user's guide for that adapter.

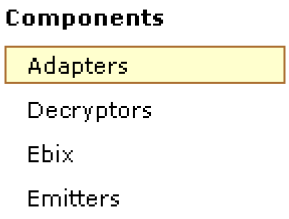
When you add an adapter (target) in the iWay Service Manager Administrator Console, the process creates run-time connection and persistent data files within iSM. In addition, the process interrogates the iSM repository entries that were built when the target and connection were created using iWay Explorer. Adding an adapter (target) also creates the run-time repository based on the design-time repository.

Adapters can be added from the iWay Service Manager Administrator Console or a process flow that is created using iWay Integration Tools (iIT) Designer.

For more information on using iIT Designer to create process flows, see the *iWay Integration Tools Designer User's Guide*.

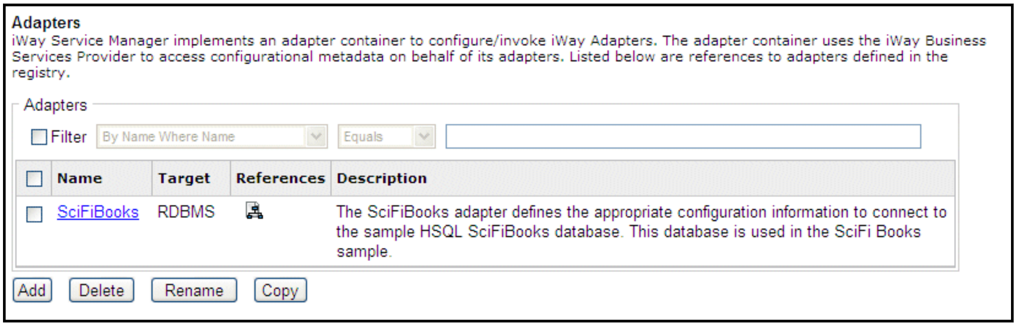
Procedure: How to Add an Adapter

To add an adapter:



1. In the left console pane of the Registry menu, select *Adapters*.

   The Adapters pane opens listing defined adapters with targets defined, as shown in the following image.



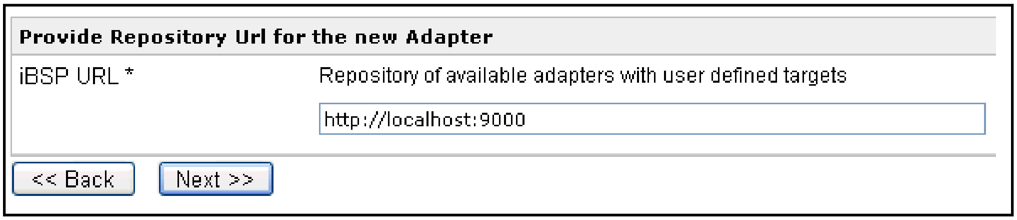
Adapters

iWay Service Manager implements an adapter container to configure/invoke iWay Adapters. The adapter container uses the iWay Business Services Provider to access configurational metadata on behalf of its adapters. Listed below are references to adapters defined in the registry.

□	Name	Target	References	Description
□	SciFiBooks	RDBMS		The SciFiBooks adapter defines the appropriate configuration information to connect to the sample HSQL SciFiBooks database. This database is used in the SciFi Books sample.

2. Click *Add*.

   The Repository pane opens, as shown in the following image.



Provide Repository Url for the new Adapter

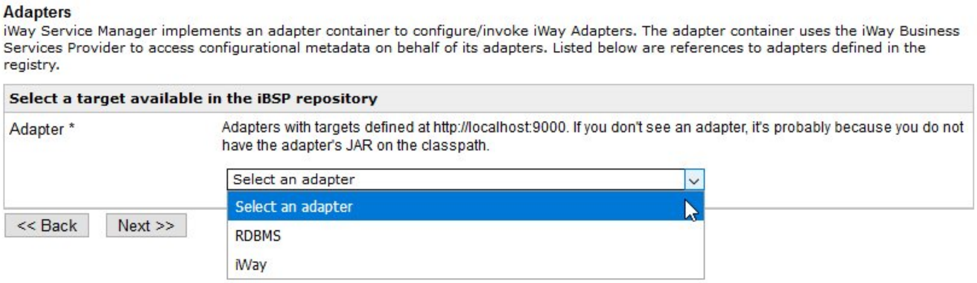
iBSP URL * Repository of available adapters with user defined targets

http://localhost:9000

3. Enter your iBSP URL, which is the location of the iWay Service Manager repository, for example, *http://localhost:9000*.

4. Click *Next*.

   An adapter selection pane opens, as shown in the following image.



Adapters

iWay Service Manager implements an adapter container to configure/invoke iWay Adapters. The adapter container uses the iWay Business Services Provider to access configurational metadata on behalf of its adapters. Listed below are references to adapters defined in the registry.

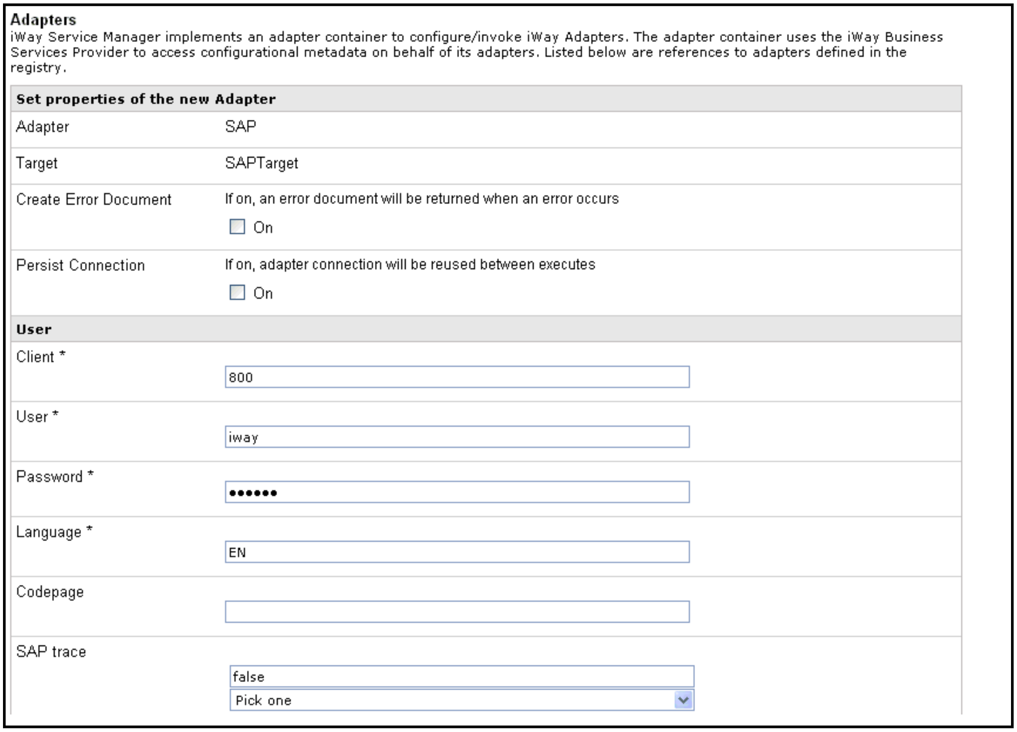
Select a target available in the iBSP repository

Adapter * Adapters with targets defined at http://localhost:9000. If you don't see an adapter, it's probably because you do not have the adapter's JAR on the classpath.

5. From the Adapter drop-down list, select an adapter, for example, SAP, then click *Next*.

6. From the Target drop-down list, select a target you configured for the adapter using iWay Explorer, then click *Next*.

The connection information associated with the target selected is displayed. For example, the following image shows the SAP connection information pane.



Adapters
iWay Service Manager implements an adapter container to configure/invoke iWay Adapters. The adapter container uses the iWay Business Services Provider to access configurational metadata on behalf of its adapters. Listed below are references to adapters defined in the registry.

Set properties of the new Adapter	
Adapter	SAP
Target	SAPTarget
Create Error Document	If on, an error document will be returned when an error occurs □ On
Persist Connection	If on, adapter connection will be reused between executes □ On
User	
Client *	800
User *	iway
Password *	••••••
Language *	EN
Codepage	
SAP trace	false Pick one

Perform the following steps:

a. Select whether to return an error document when an error occurs.

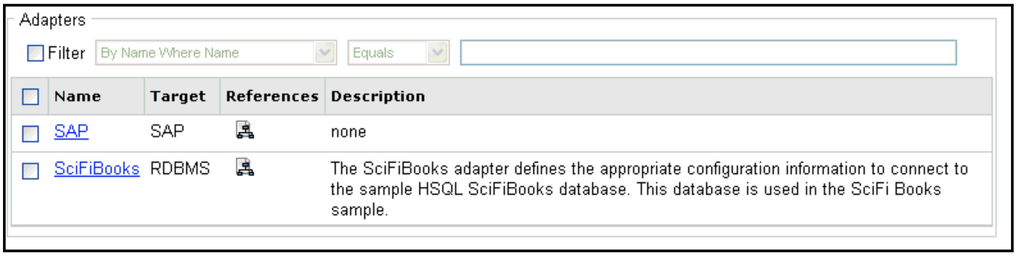
b. Select whether an adapter connection will be reused between executes.

c. Review the connection information you specified for the target in iWay Explorer. You can change or update any information.

7. Click *Next*.

8. Provide a name and, optionally, a description, for the adapter, and click *Finish*.

The adapter is added to the list in the Adapters pane, as shown in the following image.

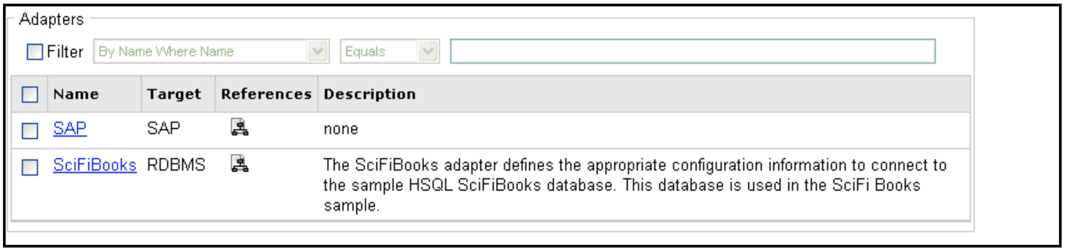


□	Name	Target	References	Description
□	SAP	SAP	🖧	none
□	SciFiBooks	RDBMS	🖧	The SciFiBooks adapter defines the appropriate configuration information to connect to the sample HSQL SciFiBooks database. This database is used in the SciFi Books sample.

Procedure: How to Modify or Update a Defined Adapter

Any adapters that have been defined using iIT Designer are available in the registry. As a result, the adapters will also appear in the Adapters pane and can be modified. For example, you can modify an adapter to include special registers (SREGs).

To modify or update a defined adapter:



□	Name	Target	References	Description
□	SAP	SAP	🖧	none
□	SciFiBooks	RDBMS	🖧	The SciFiBooks adapter defines the appropriate configuration information to connect to the sample HSQL SciFiBooks database. This database is used in the SciFi Books sample.

1. Click the name of the adapter you defined in the Adapters pane, for example, RDBMS.

   The pane that displays the target connection information opens. You cannot change the name of the adapter or the target, but you can edit the connection information.

2. After you modify the connection information, click *Update Connection Properties*.
3. After you make any changes, such as adding an RDBMS statement, or additions to the adapter target using iWay Explorer, click *Update Adapter Data*.
4. Click *Update* when you are finished.

Decryptor

Decryptors are modules that apply a decryption algorithm to the incoming message and verify the security of the message. It is used to verify whether a sender is authorized, to check that the message has not been changed, and to decrypt any part of the message that has been encrypted. Finally, the decryptor can be used to pass the message to a preparser.

Note: As of iSM version 6.1.2, using the decryptor is deprecated and not recommended.

Adding an Ebix

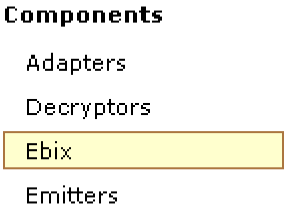
Ebix is a collection of metadata archives that define the structure of data. iWay Software provides various Ebix files to be used in conjunction with the iWay Format Adapters, such as SWIFT, X12, EDIFACT, and HIPAA.

Ebix archives are not packaged with the iWay Service Manager installation. You can download all Ebix archives from the following website, which is hosted and maintained by iWay Software:

`http://techsupport.informationbuilders.com/`

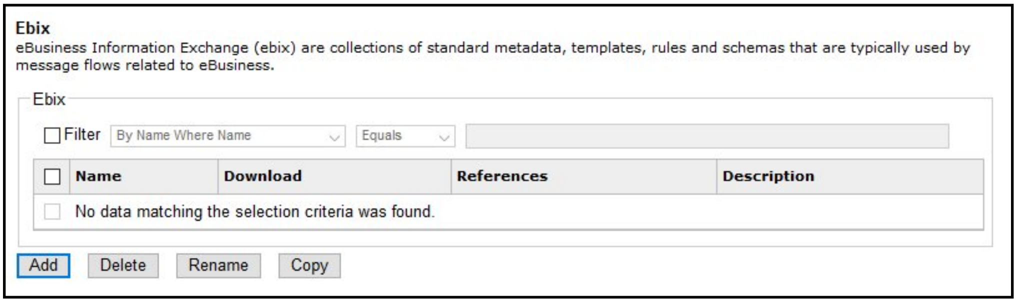
Procedure: How to Add an Ebix

To add an Ebix:



1. In the left console pane of the Registry menu, select *Ebix*.

   The Ebix pane opens.



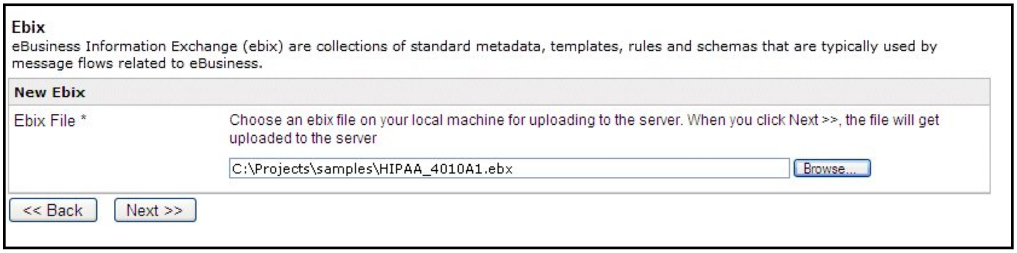
Ebix

eBusiness Information Exchange (ebix) are collections of standard metadata, templates, rules and schemas that are typically used by message flows related to eBusiness.

□	Name	Download	References	Description
□	No data matching the selection criteria was found.			

2. Click *Add*.

The New Ebix pane opens.



Ebix

eBusiness Information Exchange (ebix) are collections of standard metadata, templates, rules and schemas that are typically used by message flows related to eBusiness.

New Ebix

Ebix File *	Choose an ebix file on your local machine for uploading to the server. When you click Next >>, the file will get uploaded to the server C:\Projects\samples\HIPAA_4010A1.ebx

3. Type the path to the Ebix package on your file system or click *Browse* to find its location.

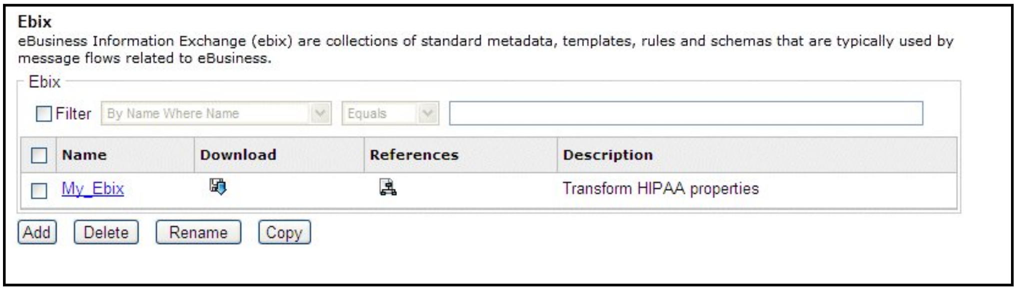
   In this example, the hipaa-package-new.ebx file is selected.

4. Click *Next*.

   The Name and Description pane opens.

5. Provide a name and, optionally, a description, for the Ebix file, and click *Finish*.

   The Ebix file is added to the list in the Ebix pane.



Ebix

eBusiness Information Exchange (ebix) are collections of standard metadata, templates, rules and schemas that are typically used by message flows related to eBusiness.

Ebix

□ Filter By Name Where Name Equals

□	Name	Download	References	Description
□	My_Ebix			Transform HIPAA properties

After an Ebix is added to iWay Service Manager, you can assign an Ebix to a channel, which is required if you want to use the Ebix. You can also use multiple Ebixes. For more information, see *Configuring Channels* on page 305.

Adding an Emitter

Emitters are transport protocols that send a document to its recipient. In addition to the emitters already defined in Service Manager, emitters are ran after the flow has completed (even when used in a process flow).

Important: It is recommended that emitters only be used and added in outlets and not process flows.

The following types of emitters can be added to iWay Service Manager:

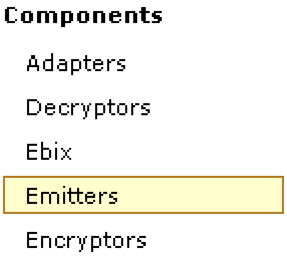
- AQ

- AS2 [nonblocking]
- Email
- File
- FTP[S} Client (Clear text or SSL FTP Clients)
- FTP[S} Client (Deprecated FTP Clients)
- HTTP 1.0 [deprecated]
- HTTP 1.1 [nonblocking] (nhttp)
- Internal Queue
- Java Message Service (jmsq)
- MQ
- MQJMS
- MSMQ
- NTCP
- Ordered Queue
- Passthru
- print
- RabitMQ
- SFTP Client (Secure Shell version FTP Client)

For more information about the emitter properties, see the *iWay Service Manager Protocol Guide*.

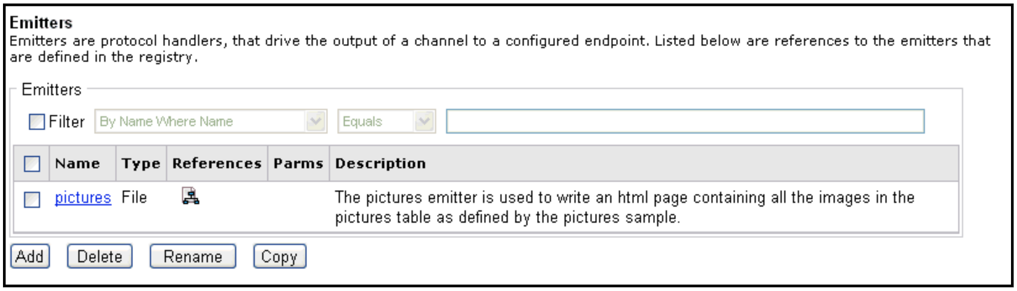
Procedure: How to Add an Emitter

To add an emitter:



1. In the left console pane of the Registry menu, select *Emitters*.

   The Emitters pane opens.



Emitters

Emitters are protocol handlers, that drive the output of a channel to a configured endpoint. Listed below are references to the emitters that are defined in the registry.

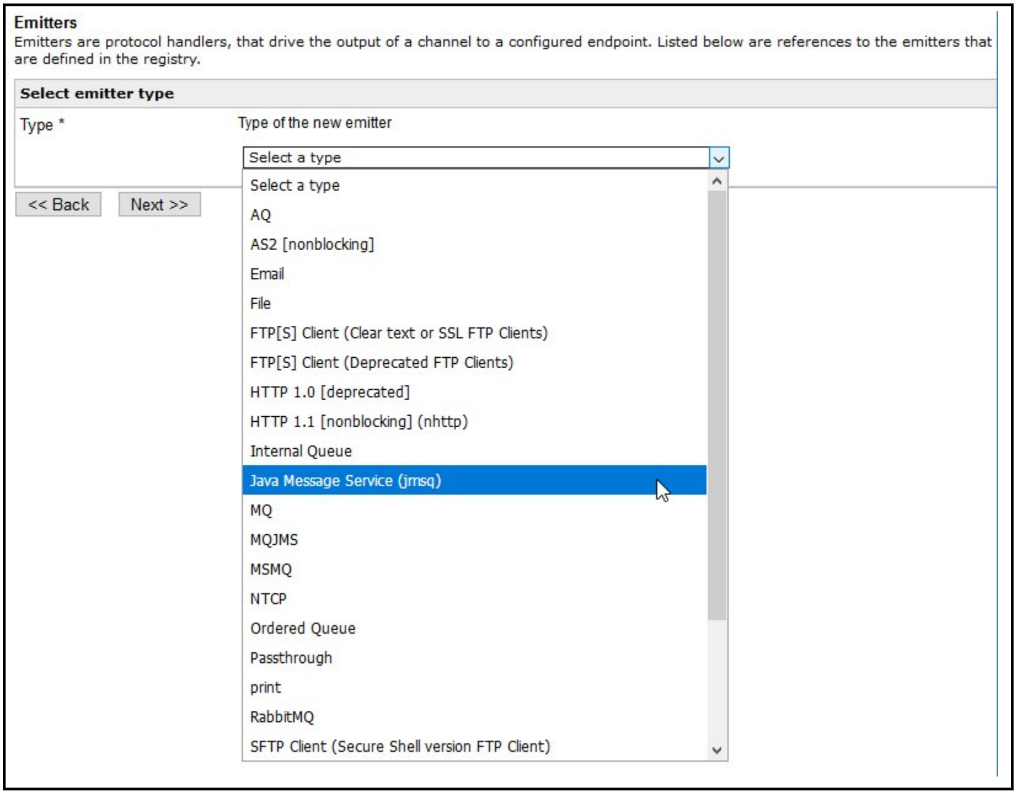
Emitters

□ Filter | By Name Where Name | Equals |

□	Name	Type	References	Parms	Description
□	pictures	File	🖧		The pictures emitter is used to write an html page containing all the images in the pictures table as defined by the pictures sample.

2. Click *Add*.

The Emitter Type pane opens.



Emitters

Emitters are protocol handlers, that drive the output of a channel to a configured endpoint. Listed below are references to the emitters that are defined in the registry.

Select emitter type

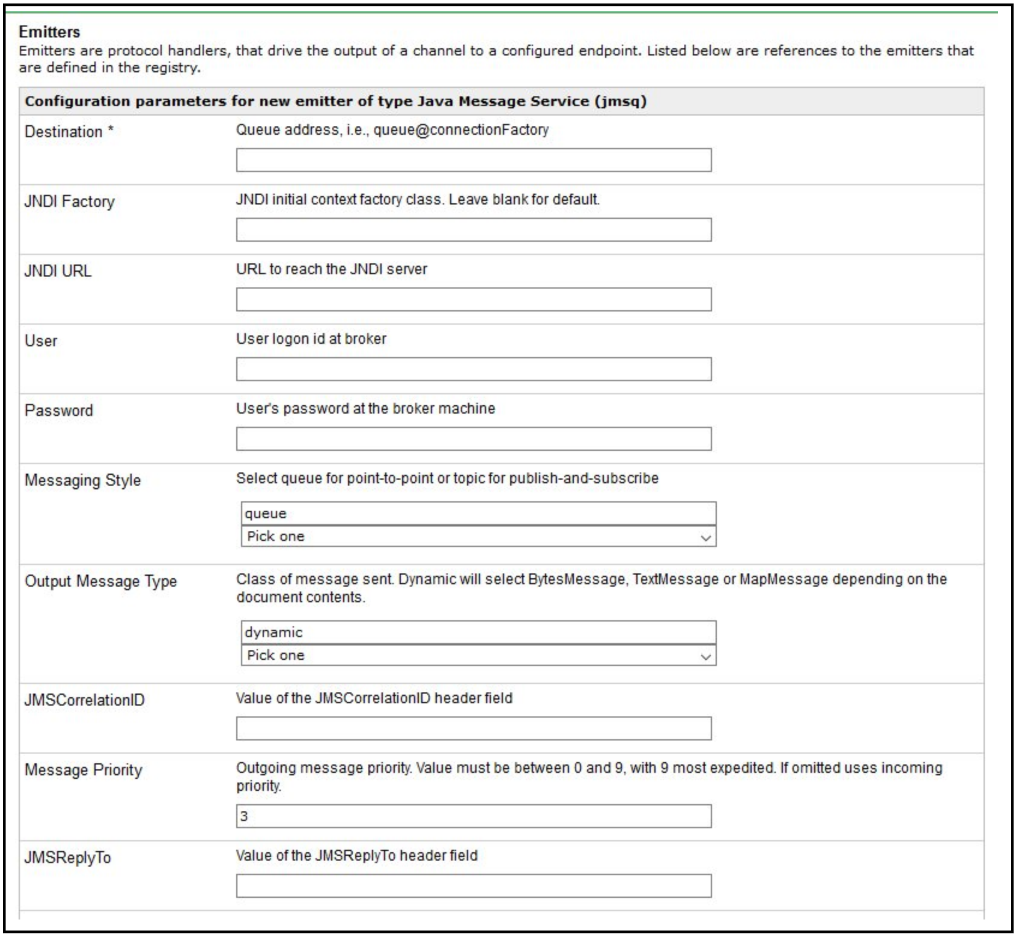
Type * — Type of the new emitter

Select a type

Select a type
AQ
AS2 [nonblocking]
Email
File
FTP[S] Client (Clear text or SSL FTP Clients)
FTP[S] Client (Deprecated FTP Clients)
HTTP 1.0 [deprecated]
HTTP 1.1 [nonblocking] (nhttp)
Internal Queue
Java Message Service (jmsq)
MQ
MQJMS
MSMQ
NTCP
Ordered Queue
Passthrough
print
RabbitMQ
SFTP Client (Secure Shell version FTP Client)
<< Back
Next >>

3. Select a type of emitter from the list of protocols, for example, JMSQ, and click *Next*.

The configuration parameters pane for the JMSQ emitter opens.



Emitters

Emitters are protocol handlers, that drive the output of a channel to a configured endpoint. Listed below are references to the emitters that are defined in the registry.

Configuration parameters for new emitter of type Java Message Service (jmsq)

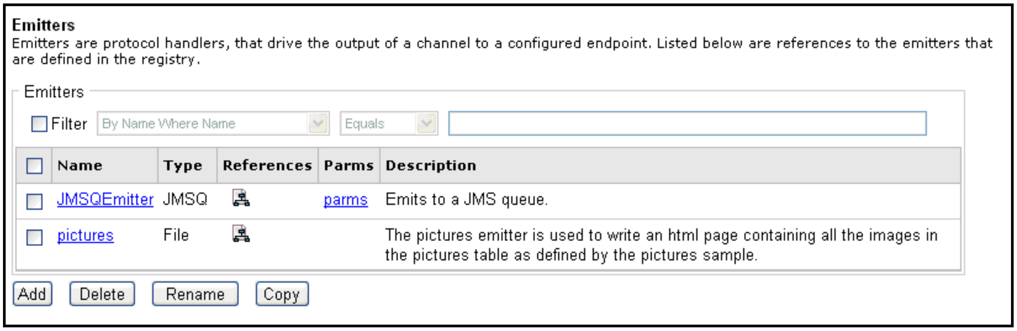
Destination *	Queue address, i.e., queue@connectionFactory	
JNDI Factory	JNDI initial context factory class. Leave blank for default.	
JNDI URL	URL to reach the JNDI server	
User	User logon id at broker	
Password	User's password at the broker machine	
Messaging Style	Select queue for point-to-point or topic for publish-and-subscribe	queue Pick one
Output Message Type	Class of message sent. Dynamic will select BytesMessage, TextMessage or MapMessage depending on the document contents.	dynamic Pick one
JMSCorrelationID	Value of the JMSCorrelationID header field	
Message Priority	Outgoing message priority. Value must be between 0 and 9, with 9 most expedited. If omitted uses incoming priority.	3
JMSReplyTo	Value of the JMSReplyTo header field	

4. Provide the required configuration parameters for the JMSQ emitter, and click *Next*.

   The Name and Description pane opens.

5. Provide a name and, optionally, a description, for the emitter, and click *Finish*.

The emitter is added to the list in the Emitters pane.



Emitters

Emitters are protocol handlers, that drive the output of a channel to a configured endpoint. Listed below are references to the emitters that are defined in the registry.

Emitters

☐ Filter: By Name Where Name | Equals

☐	Name	Type	References	Parms	Description
☐	JMSQEmitter	JMSQ	🖧	parms	Emits to a JMS queue.
☐	pictures	File	🖧		The pictures emitter is used to write an html page containing all the images in the pictures table as defined by the pictures sample.

After an emitter is added to iWay Service Manager, you can assign an emitter to an outlet that is used to construct a channel. For more information, see *Configuring Channels* on page 305.

Encryptors

Encryptors are components that are called to encrypt an outgoing document.

Note: As of iSM version 6.1.2, using the encryptor is deprecated and not recommended. Encryptors used in previous versions will continue to work, but it is recommended to revisit for redesign.

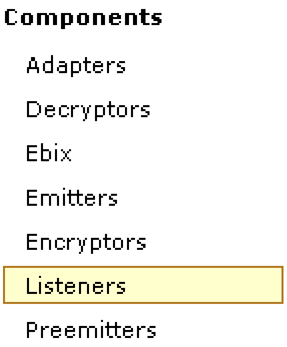
Adding a Listener

A listener is a component that receives input for a channel from a configured endpoint. Some of the types of listeners that can be employed by iSM are: AQ, AS2, Email, File, FTP (server and client), HTTP, Queuing (including MQ, JMS, MSMQ, Rabbit), SQL, SOAP and TCP (half and full duplex)

For more information on configuring listener properties, see the *iWay Service Manager Protocol Guide*.

Procedure: How to Add a Listener

To add a listener:



Components

Adapters

Decryptors

Ebix

Emitters

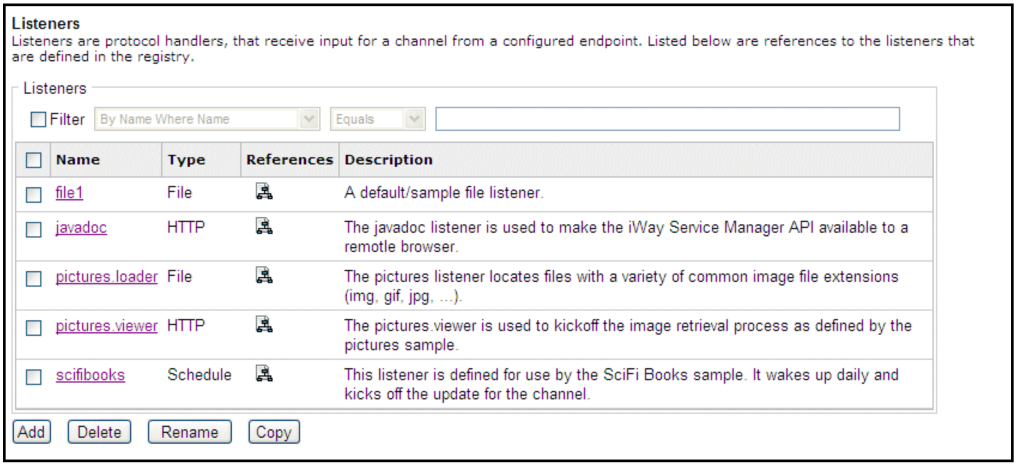
Encryptors

Listeners

Preemitters

1. In the left console pane of the Registry menu, select *Listeners*.

   The Listeners pane opens, as shown in the following image.



Listeners

Listeners are protocol handlers, that receive input for a channel from a configured endpoint. Listed below are references to the listeners that are defined in the registry.

Listeners

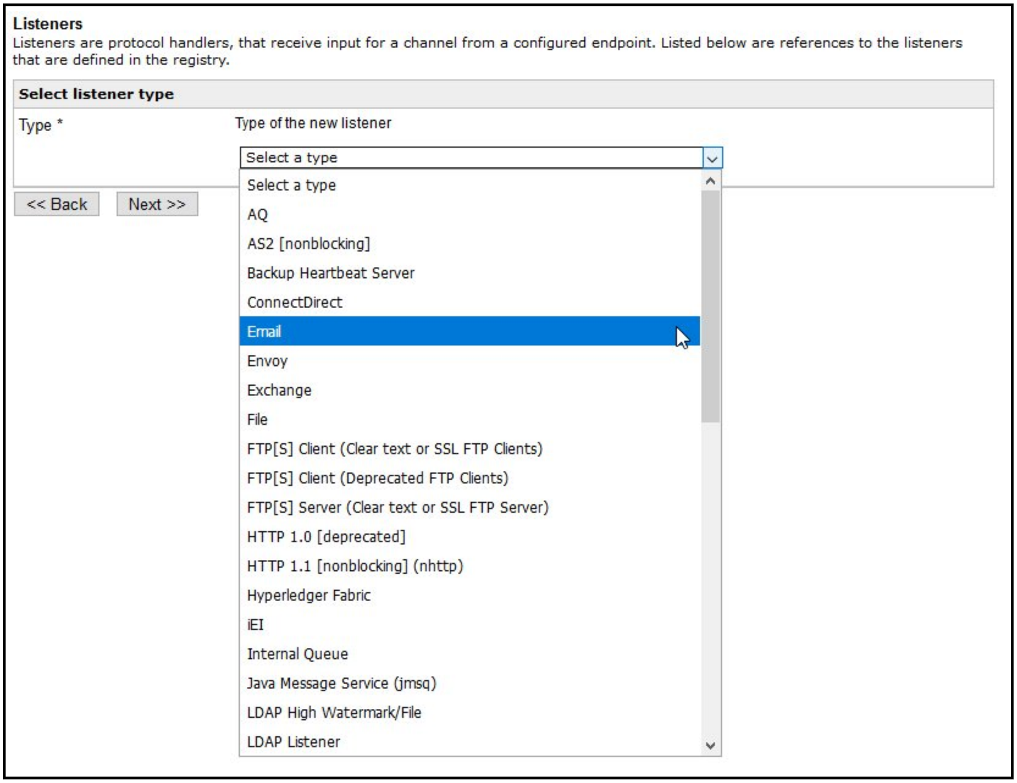
□ Filter By Name Where Name | Equals |

□	Name	Type	References	Description
□	file1	File	🖧	A default/sample file listener.
□	javadoc	HTTP	🖧	The javadoc listener is used to make the iWay Service Manager API available to a remotle browser.
□	pictures.loader	File	🖧	The pictures listener locates files with a variety of common image file extensions (img, gif, jpg, ...).
□	pictures.viewer	HTTP	🖧	The pictures.viewer is used to kickoff the image retrieval process as defined by the pictures sample.
□	scifibooks	Schedule	🖧	This listener is defined for use by the SciFi Books sample. It wakes up daily and kicks off the update for the channel.

   The table that is provided lists any existing listeners and a short description for each.

2. Click *Add*.

The Listener Type pane opens.



Listeners
Listeners are protocol handlers, that receive input for a channel from a configured endpoint. Listed below are references to the listeners that are defined in the registry.

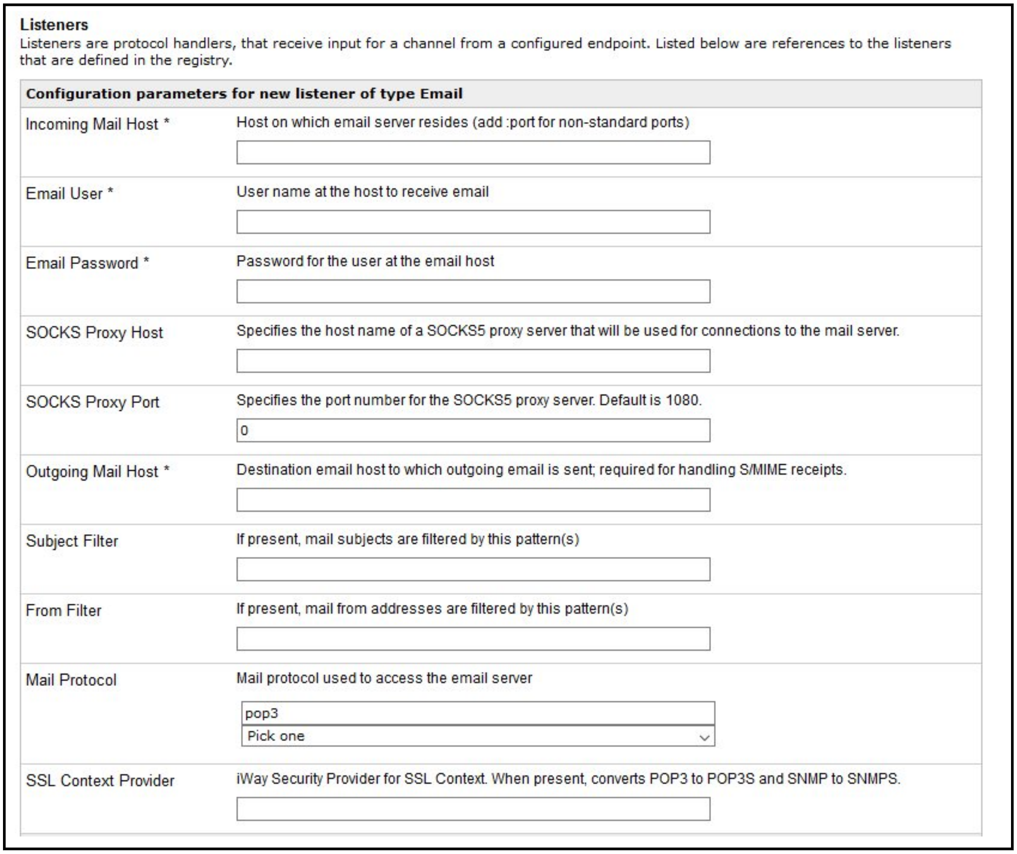
Select listener type

Type * — Type of the new listener

Select a type

3. Select a type of listener from the list, for example, EMAIL, then click *Next*.

The configuration parameters pane for the EMAIL listener opens.



Listeners
Listeners are protocol handlers, that receive input for a channel from a configured endpoint. Listed below are references to the listeners that are defined in the registry.

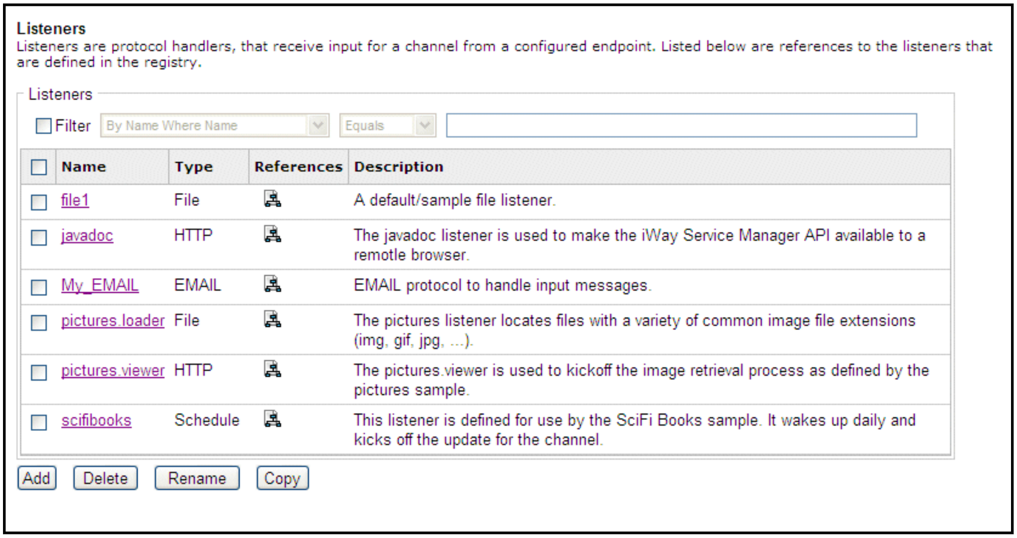
Configuration parameters for new listener of type Email	
Incoming Mail Host *	Host on which email server resides (add :port for non-standard ports)
Email User *	User name at the host to receive email
Email Password *	Password for the user at the email host
SOCKS Proxy Host	Specifies the host name of a SOCKS5 proxy server that will be used for connections to the mail server.
SOCKS Proxy Port	Specifies the port number for the SOCKS5 proxy server. Default is 1080. 0
Outgoing Mail Host *	Destination email host to which outgoing email is sent; required for handling S/MIME receipts.
Subject Filter	If present, mail subjects are filtered by this pattern(s)
From Filter	If present, mail from addresses are filtered by this pattern(s)
Mail Protocol	Mail protocol used to access the email server pop3 Pick one
SSL Context Provider	iWay Security Provider for SSL Context. When present, converts POP3 to POP3S and SNMP to SNMPS.

4. Provide the required configuration parameters for the new listener, and click *Next*.

   The Name and Description pane opens.

5. Provide a name and, optionally, a description, for the listener, and click *Finish*.

The listener is added to the list in the Listeners pane.



Listeners

Listeners are protocol handlers, that receive input for a channel from a configured endpoint. Listed below are references to the listeners that are defined in the registry.

Listeners

☐ Filter | By Name Where Name | Equals |

☐	Name	Type	References	Description
☐	file1	File	🖧	A default/sample file listener.
☐	javadoc	HTTP	🖧	The javadoc listener is used to make the iWay Service Manager API available to a remotle browser.
☐	My_EMAIL	EMAIL	🖧	EMAIL protocol to handle input messages.
☐	pictures.loader	File	🖧	The pictures listener locates files with a variety of common image file extensions (img, gif, jpg, ...).
☐	pictures.viewer	HTTP	🖧	The pictures.viewer is used to kickoff the image retrieval process as defined by the pictures sample.
☐	scifibooks	Schedule	🖧	This listener is defined for use by the SciFi Books sample. It wakes up daily and kicks off the update for the channel.

After a listener is added to iWay Service Manager, you can assign a listener to an inlet that is used to construct a channel. For more information, see *Configuring Channels* on page 305.

For more information on protocol parameters and how to define a listener, see the *iWay Service Manager Protocol Guide*.

Adding a Preemitter

A preemitter is a component that can be called just prior to sending an output document. Normally it is used to convert a payload document in internal format into an external format such as EDI or CSV.

Each preemitter uses a class file that must be located in a directory in the Java classpath. iWay Service Manager includes preemitters that have been preconfigured. The following preconfigured preemitters are available to be added to iWay Service Manager:

- Constant PE (com.ibi.preemit.XDConstantPE)
- Deflate (com.ibi.preemit.XDDeflate)
- File read (com.ibi.preemit.XDFilePreEmitter)
- iWay Transformations (com.ibi.preemit.XDXMLGpreEmitter)
- Legacy Record Preemitter (com.ibi.preemit.LegacyRecordPreemitter)

- Marshalls a Message (com.ibi.preemit.Marshalls)
- New SMIMEPE (com.ibi.preemit.XDNewSMIMEPE)
- PGP Encryption and Signature (com.ibi.preemit.PGPEncrypt)
- PGP Sign Only (com.ibi.preemit.XDPGPSignEmitter)
- QA Agent (com.ibi.preemit.XDQAPrint)
- Remove outer tag (com.ibi.preemit.XDDeTag)
- Replace characters on a one for one basis (com.ibi.preemit.XDCharRepl)
- XDSWIFTPreEmitter (com.ibi.preemit.XDSwiftpreEmitter)
- XML Entity Character Mapper (com.ibi.preemit.XDEntityRepl)
- XML Input to Flat Value (com.ibi.preemit.XDFlatDelimPreEmitter)
- XSLT Pre Emitter (com.ibi.preemit.XDXSLTpreEmitter)

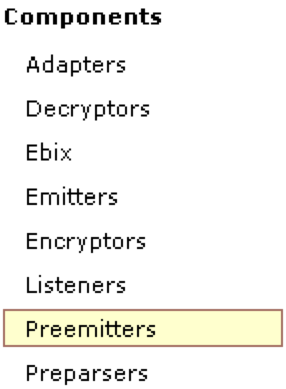
Note: Preemitters can be chained.

For more information about configuring specific preemitter properties, see the *iWay Service Manager Component and Functional Language Reference Guide*.

For more information on the methodology used in writing preemitters, see the *iWay Service Manager Programmer's Guide.*

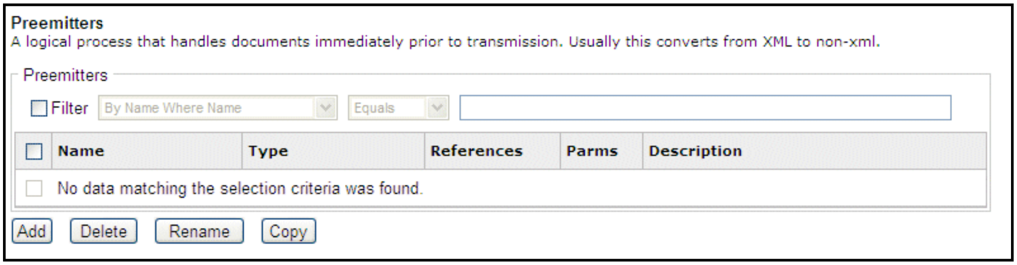
Procedure: How to Add a Preemitter

To add a preemitter:



1. In the left console pane of the Registry menu, select *Preemitters*.

The Preemitters pane opens.



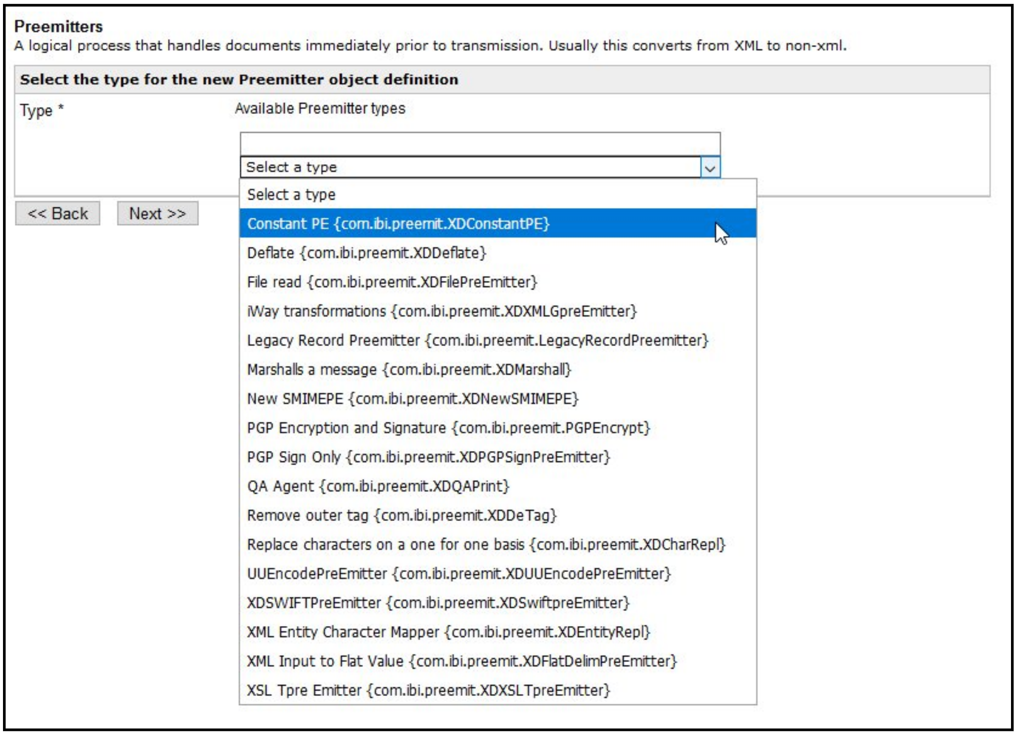
Preemitters

A logical process that handles documents immediately prior to transmission. Usually this converts from XML to non-xml.

□	Name	Type	References	Parms	Description
□	No data matching the selection criteria was found.				

2. Click *Add*.

   The Preemitter Type pane opens.



Preemitters

A logical process that handles documents immediately prior to transmission. Usually this converts from XML to non-xml.

Select the type for the new Preemitter object definition

Type * Available Preemitter types

Select a type

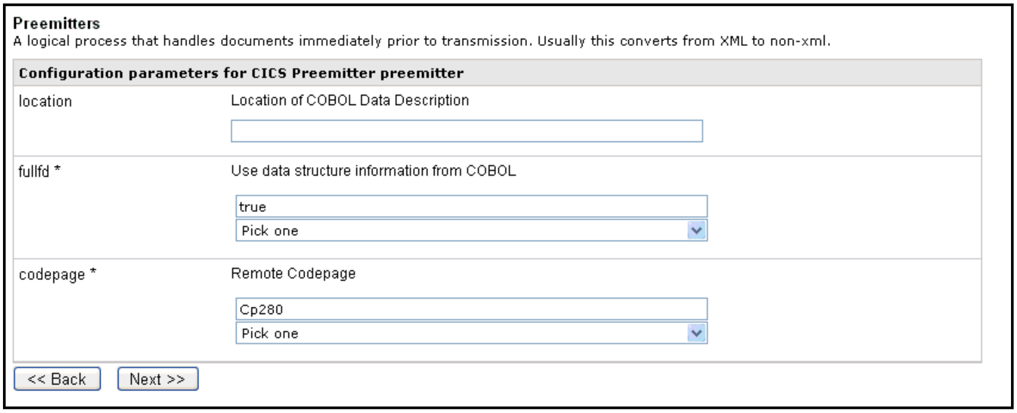
Select a type
Constant PE {com.ibi.preemit.XDConstantPE}
Deflate {com.ibi.preemit.XDDeflate}
File read {com.ibi.preemit.XDFilePreEmitter}
iWay transformations {com.ibi.preemit.XDXMLGpreEmitter}
Legacy Record Preemitter {com.ibi.preemit.LegacyRecordPreemitter}
Marshalls a message {com.ibi.preemit.XDMarshall}
New SMIMEPE {com.ibi.preemit.XDNewSMIMEPE}
PGP Encryption and Signature {com.ibi.preemit.PGPEncrypt}
PGP Sign Only {com.ibi.preemit.XDPGPSignPreEmitter}
QA Agent {com.ibi.preemit.XDQAPrint}
Remove outer tag {com.ibi.preemit.XDDeTag}
Replace characters on a one for one basis {com.ibi.preemit.XDCharRepl}
UUEncodePreEmitter {com.ibi.preemit.XDUUEncodePreEmitter}
XDSWIFTPreEmitter {com.ibi.preemit.XDSwiftpreEmitter}
XML Entity Character Mapper {com.ibi.preemit.XDEntityRepl}
XML Input to Flat Value {com.ibi.preemit.XDFlatDelimPreEmitter}
XSL Tpre Emitter {com.ibi.preemit.XDXSLTpreEmitter}
<< Back
Next >>

3. Select a type of preemitter from the list, for example, CICS Preemitter (com.ibi.preemit.CICSPreemitter).

   You can also manually type the class name of a preemitter that is packaged in a .jar file, which is in the CLASSPATH.

4. Click *Next*.

The configuration parameters pane for the CICS preemitter opens.



Preemitters

A logical process that handles documents immediately prior to transmission. Usually this converts from XML to non-xml.

Configuration parameters for CICS Preemitter preemitter

location	Location of COBOL Data Description	
fullfd *	Use data structure information from COBOL	true
codepage *	Remote Codepage	Cp280

5. Provide the required configuration parameters for the preemitter, and click *Next*.

   The Name and Description pane opens.

6. Provide a name and, optionally, a description, for the preemitter, and click *Finish*.

   The preemitter is added to the list in the Preemitters pane.

   After a preemitter is added to iWay Service Manager, you can assign a preemitter to an outlet that is used to construct a channel. For more information, see *Configuring Channels* on page 305.

Adding a Preparser

A preparser is designed to convert incoming messages into processable documents. The preparsed document then passes through the standard transformation services to reach the designated processing service. An example of a preparser is a class that accepts an EDI-formatted document and converts it to XML for further processing. For more information on the methodology used in writing preparsers, see the *iWay Service Manager Programmer's Guide*.

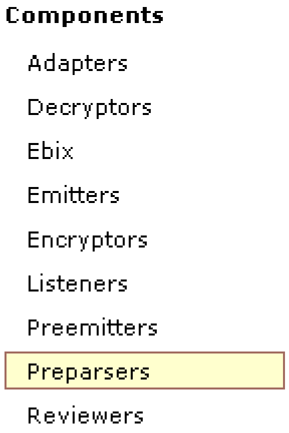
Each preparser uses a class file that must be located in a directory which is in the Java classpath. iWay Service Manager includes preparsers that have been preconfigured.

A Service Manager flow can include multiple preparsers and more than one preparser can handle a document. For example, a document can be processed by preparser A and then, subsequently, by preparser B.

Important: A preparser is run prior to a process flow. Further message processing can be handled in an iWay process flow.

Procedure: How to Add a Preparser

To add a preparser:

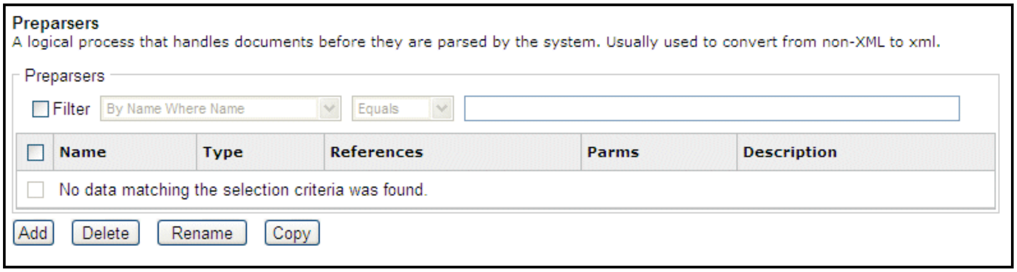


Components

- Adapters
- Decryptors
- Ebix
- Emitters
- Encryptors
- Listeners
- Preemitters
- Preparsers
- Reviewers

1. In the left console pane of the Registry menu, select *Preparsers*.

   The Preparsers pane opens.



Preparsers

A logical process that handles documents before they are parsed by the system. Usually used to convert from non-XML to xml.

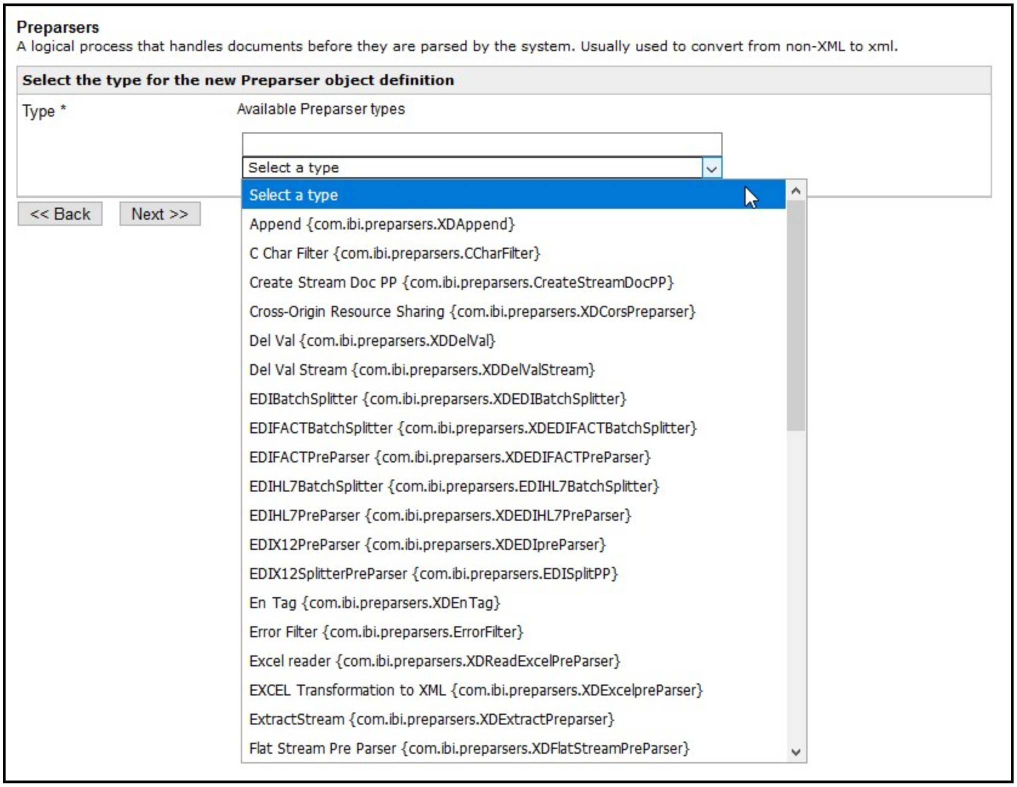
Preparsers

☐ Filter By Name Where Name Equals

☐	Name	Type	References	Parms	Description
☐	No data matching the selection criteria was found.				

2. Click *Add*.

The Preparser Type pane opens.



Preparsers

A logical process that handles documents before they are parsed by the system. Usually used to convert from non-XML to xml.

Select the type for the new Preparser object definition

Type * Available Preparser types

Select a type

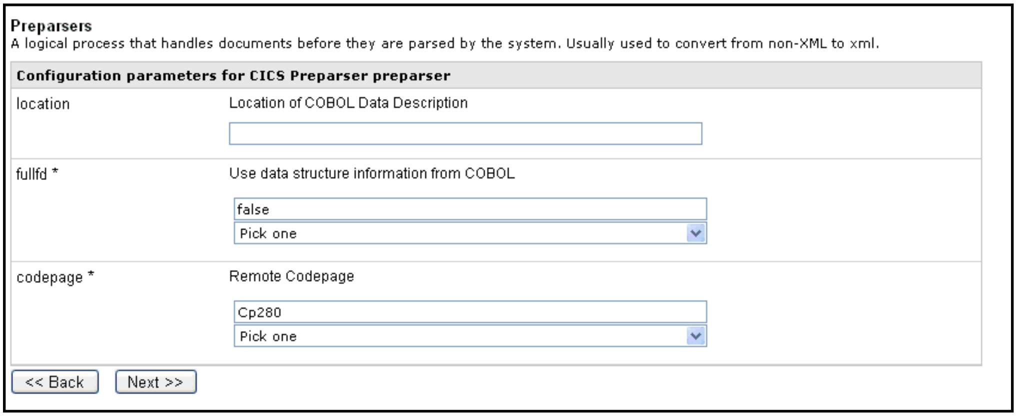
- Select a type
- Append {com.ibi.preparsers.XDAppend}
- C Char Filter {com.ibi.preparsers.CCharFilter}
- Create Stream Doc PP {com.ibi.preparsers.CreateStreamDocPP}
- Cross-Origin Resource Sharing {com.ibi.preparsers.XDCorsPreparser}
- Del Val {com.ibi.preparsers.XDDelVal}
- Del Val Stream {com.ibi.preparsers.XDDelValStream}
- EDIBatchSplitter {com.ibi.preparsers.XDEDIBatchSplitter}
- EDIFACTBatchSplitter {com.ibi.preparsers.XDEDIFACTBatchSplitter}
- EDIFACTPreParser {com.ibi.preparsers.XDEDIFACTPreParser}
- EDIHL7BatchSplitter {com.ibi.preparsers.EDIHL7BatchSplitter}
- EDIHL7PreParser {com.ibi.preparsers.XDEDIHL7PreParser}
- EDIX12PreParser {com.ibi.preparsers.XDEDIpreParser}
- EDIX12SplitterPreParser {com.ibi.preparsers.EDISplitPP}
- En Tag {com.ibi.preparsers.XDEnTag}
- Error Filter {com.ibi.preparsers.ErrorFilter}
- Excel reader {com.ibi.preparsers.XDReadExcelPreParser}
- EXCEL Transformation to XML {com.ibi.preparsers.XDExcelpreParser}
- ExtractStream {com.ibi.preparsers.XDExtractPreparser}
- Flat Stream Pre Parser {com.ibi.preparsers.XDFlatStreamPreParser}

3. Select a type of preparser from the list, for example, CICS Preparser (com.ibi.preparsers.CICSPreparser).

   You can also manually type the class name of a preparser that is packaged in a .jar file, which is in the CLASSPATH.

4. Click *Next*.

The configuration parameters pane for the CICS preparser opens.



Preparsers

A logical process that handles documents before they are parsed by the system. Usually used to convert from non-XML to xml.

Configuration parameters for CICS Preparser preparser

location	Location of COBOL Data Description	
fullfd *	Use data structure information from COBOL	false
codepage *	Remote Codepage	Cp280

5. Provide the required configuration parameters for the preparser, and click *Next*.

   The Name and Description pane opens.

6. Provide a name and, optionally, a description, for the preparser, and click *Finish*.

   The preparser is added to the list in the Preparsers pane.

   After a preparser is added to iWay Service Manager, you can assign a preparser to an inlet that is used to construct a channel. For more information, see *Configuring Channels* on page 305.

Streaming Preparser

To handle large files, iWay Service Manager provides a streaming interface. You can use the streaming preparser only with the HTTP, File, FTP, Sonic, SFTP, nHTTP, MQ, and JMS listeners. To use a streaming preparser in conjunction with other preparsers, you must define it as the first preparser in the chain.

Examples of available streaming preparsers are:

❏ Line-Oriented Preparser

❏ Concatenated XML Document Preparser

❏ XML Splitter Preparser

Line-Oriented Preparser

iWay Service Manager includes a line-oriented preparser designed to accept an input stream and create smaller input files from the original large file. You can configure the preparser to specify the character that indicates a new record.

The following incoming document can be subdivided into three individual documents, based on the delimiter:

```
What is your name?
Where do you live?
When were you born?
```

This document can be divided into three individual documents.

```
What is your name?
```

```
Where do you live?
```

```
When were you born?
```

The preparsers pass the individual documents to the remainder of the flow. The transaction extends for all records in the stream, as per standard iWay Service Manager transaction control. Each record, however, is independent for processing purposes.

Procedure: How to Add a Line-Oriented Preparser (Flat Stream PreParser)

To define a line-oriented preparser:

1. In the left console pane of the Registry menu, select *Preparsers*.

   The Preparsers pane opens.

2. Click *Add*.
   The Preparser Type pane opens.

3. In the Type field, select *Flat Stream PreParser {com.ibi.preparsers.XDFlatStreamPreParser}*.

4. Click *Next*.

   The configuration parameters pane for Flat Stream preparser opens.

5. In the Delimiters field, type a value that indicates possible delimiters. This parameter is optional.

6. Click *Next*.

7. Provide a name and a description (optional) for the preparser, and click *Finish*.

   The preparser is added to the list in the Preparsers pane.

Reference: Flat Stream PreParser

The Flat Stream PreParser splits non-XML messages on a recognized character. The configuration specifies the split character. The default is the end of line for the platform. The character can be specified as any character, a special character, or a hexadecimal character.

A standard use of this preparser is to handle individual lines of a large, delimited file. The following table lists and describes the character representations.

Character Representation	Description
X	Any character
\n	New line
\t	Tab
\r	Carriage return
\xab	Ab are hexadecimal characters such as 0A.

Concatenated XML Document Preparser

iWay Service Manager includes a preparser designed to accept input files containing multiple XML documents and create single XML documents from the original input file.

The preparser processes the following batch of XML documents:

```
<?xml version="1.0" encoding="UTF-8"?>
<hi>This is my document</hi>
<hi>This is my document</hi>
<hi>This is my document</hi>
<hi>This is my document</hi>
```

Procedure: How to Add a Concatenated XML Document Preparser

To define a concatenated XML document preparser:

1. In the left console pane of the Registry menu, select *Preparsers*.

   The Preparsers pane opens.

2. Click *Add*.
   The Preparser Type pane opens.

3. In the Type field, select *XML Stream PreParser {com.ibi.preparsers.XDXMLStreamPreParser}* and click *Next*.

4. Provide a name and a description (optional) for the preparser, and click *Finish*.

   The preparser is added to the list in the Preparsers pane.

Reference: XML Stream PreParser

The incoming XML document is divided into smaller documents, broken at a specified repeating child, for example:

```
<a>
<b>
<c>one</c>
<b>
<b>
<c>two</c>
</b>
</a>
```

With a split specified as a/b, it results in three passes through the configured flow.

Pass 1

```
<a>
<b>
<c>one</c>
</b>
</a>
```

Pass 2

```
<a>
<b>
<c>two</c>
</b>
</a>
```

Pass 3

No data.

XML Splitter Preparser

iWay Service Manager includes a tag-oriented XML splitter preparser designed to accept an input XML document and create smaller XML files from the original large file. You can configure the preparser to specify the tags or combination of tags to use to split the document.

The preparser supports the last level of the XML splitter preparser as the wildcard, *. This feature causes each entry below the fixed levels to be split off from the input XML document.

The following incoming document with the split parameters of A,B,* can be subdivided into two individual documents.

Incoming Document	Individual Document 1	Individual Document 2

`<A>`
`<B>`
`<C1/>`
`<C2>`
`<D1/>`
`</C2>`
`</B>`
`</A>`
	`<A>`
	`<B>`
	`</B>`
	`</A>`
		`<A>`
		`<B>`
		`<C2>`
		`<D1/>`
		`</C2>`
		`</B>`
		`</A>`

Procedure: How to Add an XML Splitter Preparser

To add an XML splitter preparser:

1. In the console, select *Registry*, and then *Preparsers*.

   The Preparsers pane opens displaying a list of configured preparsers. You can modify these preparsers or add a new one.

2. Click *Add*.
   The New Preparser Type pane opens.

3. In the Type field, select *XML Split {com.ibi.preparsers.XDXMLSplit}*.

4. Click *Next*.

   The Configuration Parameters pane opens.

   a. In the Level String field, type the path to the element on which to split. For this example, the path would be entered as:

   ```
   /ConfirmBOD
   ```

   b. Select *True* in the Sequence drop-down to set a count attribute in the root.

5. Click *Next*.

6. Provide a name and a description (optional) for the preparser, and click *Finish*.

   The Preparsers pane opens with your new preparser added to the list.

Example: Adding an XML Splitter Preparser

The following incoming document must be split into two smaller documents:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Root>
      <ConfirmBOD lang="en-US">
            <ApplicationArea>
                  <Sender>
                       <LogicalId>3000</LogicalId>
                  </Sender>
            </ApplicationArea>
            <DataArea>
                  <Confirm>Failure</Confirm>
            </DataArea>
      </ConfirmBOD>
      <ConfirmBOD lang="en-US">
            <ApplicationArea>
                  <Sender>
                       <LogicalId>4000</LogicalId>
                  </Sender>
            </ApplicationArea>
            <DataArea>
                  <Confirm>Success</Confirm>
            </DataArea>
      </ConfirmBOD>
</Root>
```

The document must be split on the ConfirmBOD tag. The preparser should act on the document and create two documents that look similar to the following:

```
<Root>
      <ConfirmBOD lang=...>
         ....
      </ConfirmBOD>
</Root>
```

Adding a Reviewer

There are two types of reviewers. An **inbound** reviewer is the first exit to receive a document after parsing. An **outbound** reviewer is the last exit to receive the document prior to the actual emit operation. These exits are intended for envelope handling but can be used for any desired purpose.

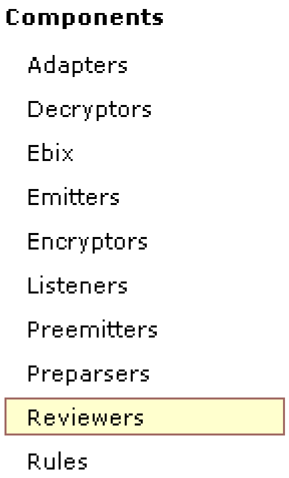
Reviewers receive an incoming document as input, and are responsible for loading the outbound document with the appropriate information after the review. For example, an inbound reviewer might handle the WS-SECURITY header, operating on the payload of the document as directed by the fields in the header.

If more than one reviewer is defined for a message, they are chained, such that the output of one reviewer is the input to the next. Several reviewers, one following the other, might each be responsible for handling one type of envelope header.

For more information on reviewers, see the *iWay Service Manager Programmer's Guide*.

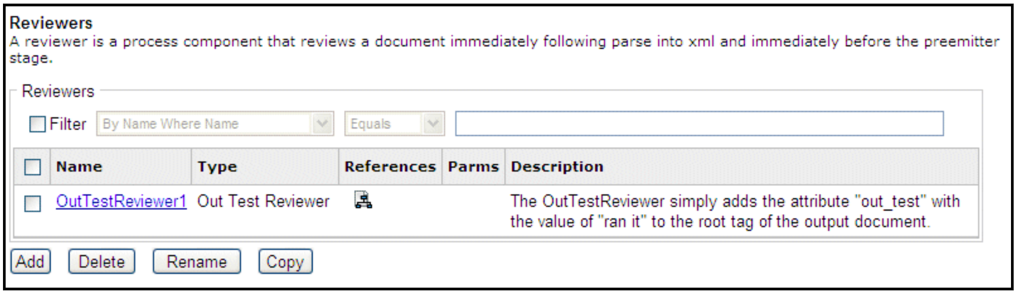
Procedure: How to Add a Reviewer

To add a reviewer:



1. In the left console pane of the Registry menu, select *Reviewers*.

   The Reviewers pane opens.



Reviewers

A reviewer is a process component that reviews a document immediately following parse into xml and immediately before the preemitter stage.

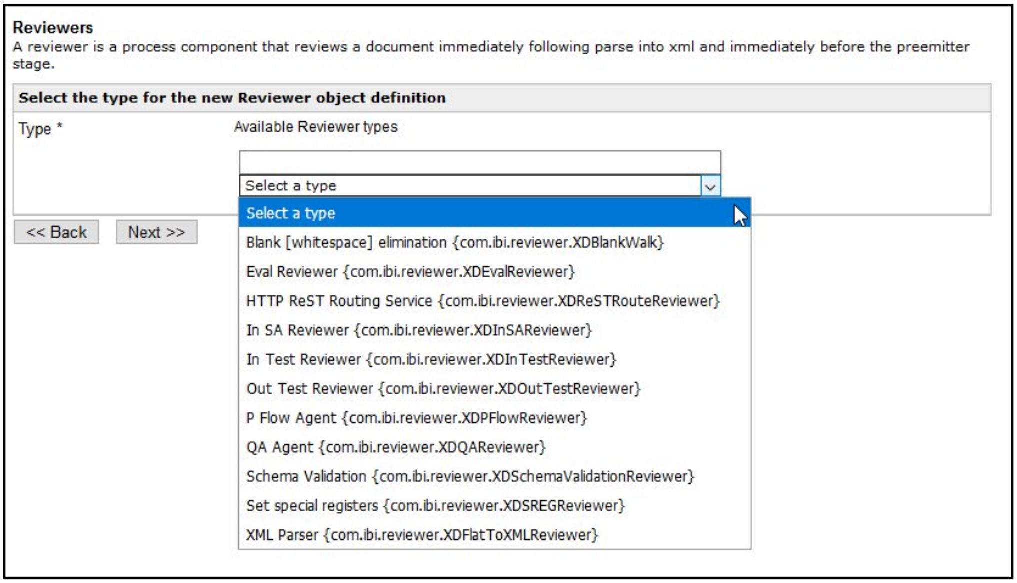
Reviewers

□ Filter By Name Where Name Equals

□	Name	Type	References	Parms	Description
□	OutTestReviewer1	Out Test Reviewer			The OutTestReviewer simply adds the attribute "out_test" with the value of "ran it" to the root tag of the output document.

2. Click *Add*.

The Reviewer Type pane opens.



Reviewers

A reviewer is a process component that reviews a document immediately following parse into xml and immediately before the preemitter stage.

Select the type for the new Reviewer object definition

Type * Available Reviewer types

Select a type

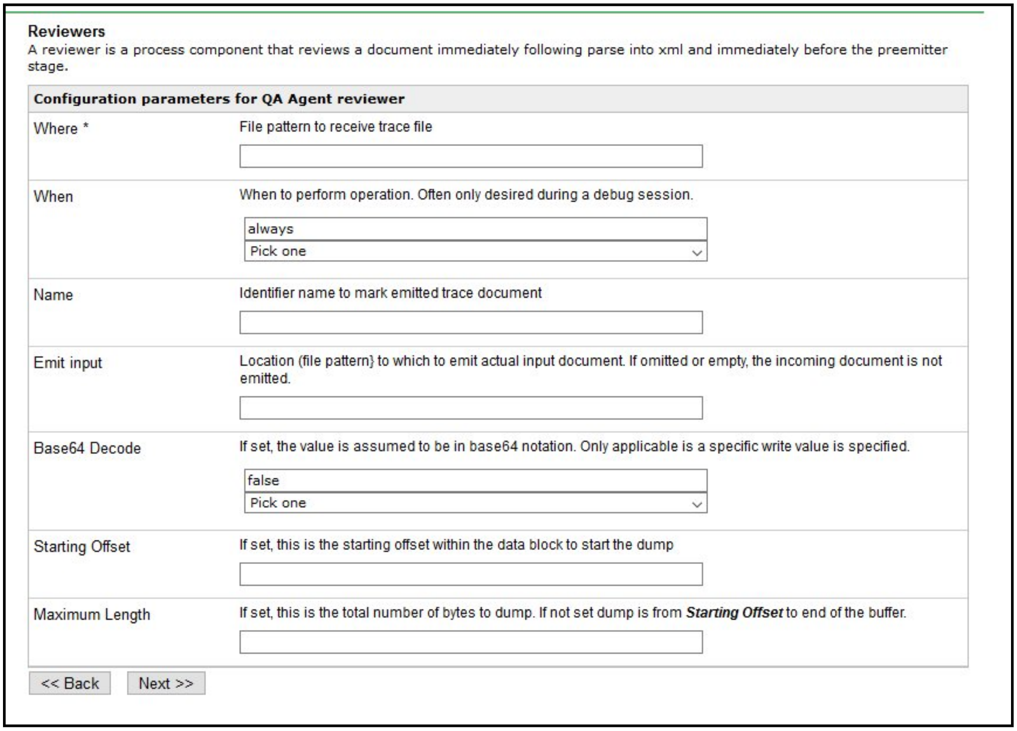
- Select a type
- Blank [whitespace] elimination {com.ibi.reviewer.XDBlankWalk}
- Eval Reviewer {com.ibi.reviewer.XDEvalReviewer}
- HTTP ReST Routing Service {com.ibi.reviewer.XDReSTRouteReviewer}
- In SA Reviewer {com.ibi.reviewer.XDInSAReviewer}
- In Test Reviewer {com.ibi.reviewer.XDInTestReviewer}
- Out Test Reviewer {com.ibi.reviewer.XDOutTestReviewer}
- P Flow Agent {com.ibi.reviewer.XDPFlowReviewer}
- QA Agent {com.ibi.reviewer.XDQAReviewer}
- Schema Validation {com.ibi.reviewer.XDSchemaValidationReviewer}
- Set special registers {com.ibi.reviewer.XDSREGReviewer}
- XML Parser {com.ibi.reviewer.XDFlatToXMLReviewer}

3. Select a type of reviewer from the list, for example, QA Agent.

   You can also manually type the class name of a reviewer that is packaged in a .jar file, which is in the CLASSPATH.

4. Click *Next*.

The configuration parameters pane for the reviewer opens.



Reviewers

A reviewer is a process component that reviews a document immediately following parse into xml and immediately before the preemitter stage.

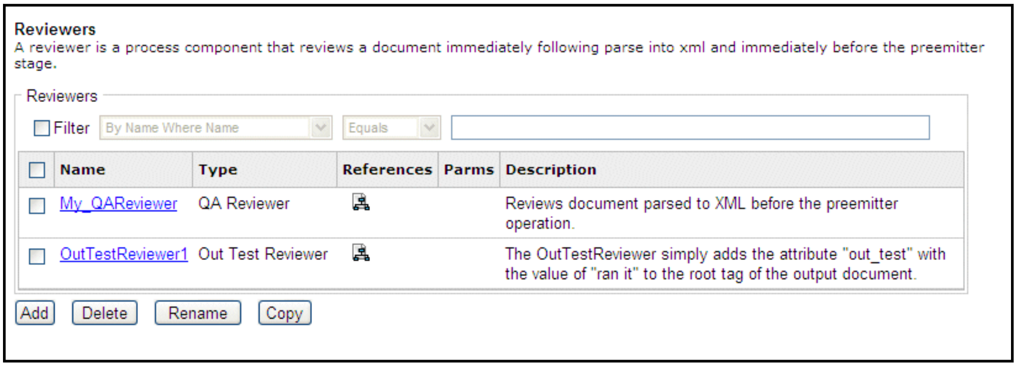
Configuration parameters for QA Agent reviewer		
Where *	File pattern to receive trace file	
When	When to perform operation. Often only desired during a debug session.	always Pick one
Name	Identifier name to mark emitted trace document	
Emit input	Location (file pattern) to which to emit actual input document. If omitted or empty, the incoming document is not emitted.	
Base64 Decode	If set, the value is assumed to be in base64 notation. Only applicable is a specific write value is specified.	false Pick one
Starting Offset	If set, this is the starting offset within the data block to start the dump	
Maximum Length	If set, this is the total number of bytes to dump. If not set dump is from ***Starting Offset*** to end of the buffer.	

5. Provide the required configuration parameters for the reviewer, and click *Next*.

   The Name and Description pane opens.

6. Provide a name and, optionally, a description, for the reviewer, and click *Finish*.

   The reviewer is added to the list in the Reviewers pane.



Reviewers

A reviewer is a process component that reviews a document immediately following parse into xml and immediately before the preemitter stage.

Reviewers

□ Filter By Name Where Name | Equals |

□	Name	Type	References	Parms	Description
□	My_QAReviewer	QA Reviewer	🖧		Reviews document parsed to XML before the preemitter operation.
□	OutTestReviewer1	Out Test Reviewer	🖧		The OutTestReviewer simply adds the attribute "out_test" with the value of "ran it" to the root tag of the output document.

After a reviewer is added to iWay Service Manager, you can add it to an outlet that is used in a channel.

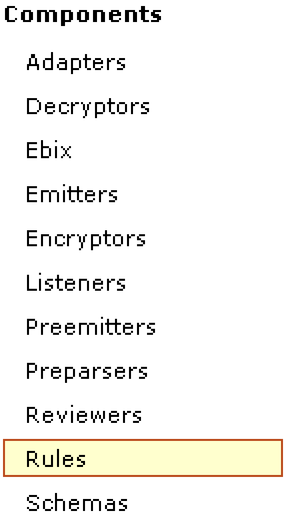
Adding a Rule

iWay Service Manager provides functionality to validate the structure and content of incoming and outgoing messages/documents. This process goes beyond simply checking the structure that may be expressed in a schema. Rules allow you to validate content values, complex conditional dependencies of elements, and balance values.

Note: Rules in an ebix file do not need to be added; they are automatically discovered. The rules added following the procedure below are used for SWIFT network rules and custom rules only.

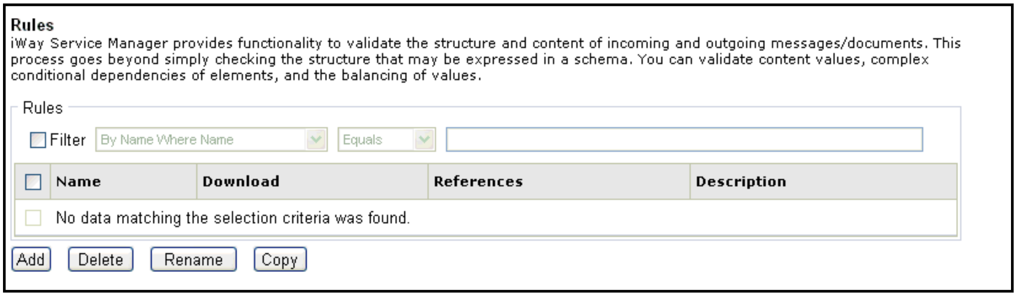
Procedure: How to Add a Rule

To add a rule:



1. In the left console pane of the Registry menu, select *Rules*.

The Rules pane opens.



Rules

iWay Service Manager provides functionality to validate the structure and content of incoming and outgoing messages/documents. This process goes beyond simply checking the structure that may be expressed in a schema. You can validate content values, complex conditional dependencies of elements, and the balancing of values.

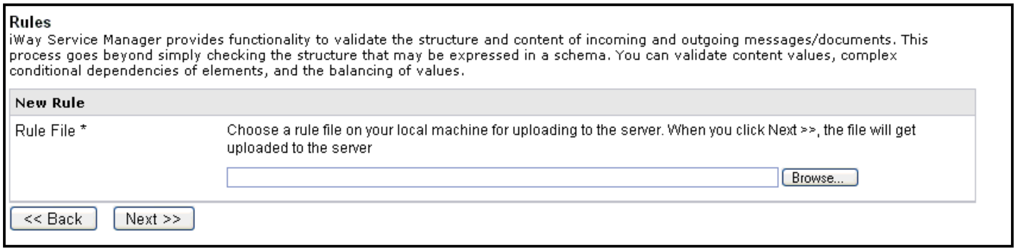
Rules

□ Filter By Name Where Name Equals

□	Name	Download	References	Description
□	No data matching the selection criteria was found.			

2. Click *Add*.

   The New Rule pane opens.



Rules

iWay Service Manager provides functionality to validate the structure and content of incoming and outgoing messages/documents. This process goes beyond simply checking the structure that may be expressed in a schema. You can validate content values, complex conditional dependencies of elements, and the balancing of values.

New Rule

Rule File * Choose a rule file on your local machine for uploading to the server. When you click Next >>, the file will get uploaded to the server

3. Type the path to the rule file on your file system or click *Browse* to find its location.
4. Click *Next*.

   The Name and Description pane opens.

5. Provide a name and, optionally, a description, for the rule file, and click *Finish*.

   The rule file is uploaded to the server and is added to the list in the Rules pane.

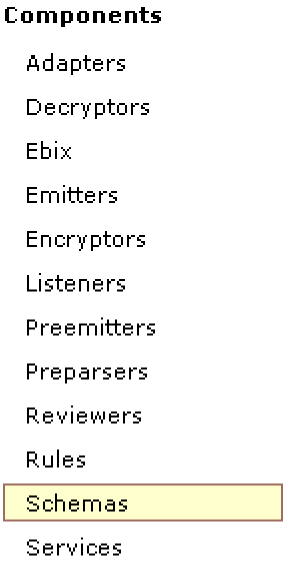
   For more information on rules files, see the *iWay Service Manager Programmer's Guide*.

Adding a Schema

The XML Schema Definition (XSD) language enables you to define the structure and data types for XML documents and data. An XML schema defines the elements, attributes, and data types that conform to the World Wide Web Consortium (W3C) standard. Schemas are primarily used for design time by process flows and transforms. However, there are runtime components that use schemas at runtime to validate the document.

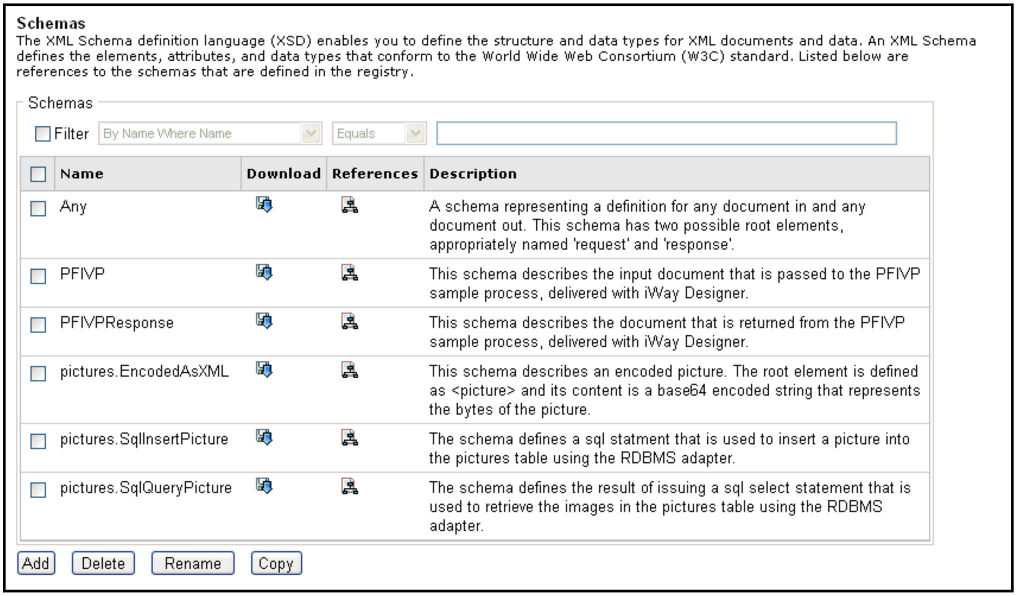
Procedure: How to Add a Schema

To add a schema:



1. In the left console pane of the Registry menu, select *Schemas*.

The Schemas pane opens. Schemas already defined in Service Manager and iIT Designer are listed.



Schemas

The XML Schema definition language (XSD) enables you to define the structure and data types for XML documents and data. An XML Schema defines the elements, attributes, and data types that conform to the World Wide Web Consortium (W3C) standard. Listed below are references to the schemas that are defined in the registry.

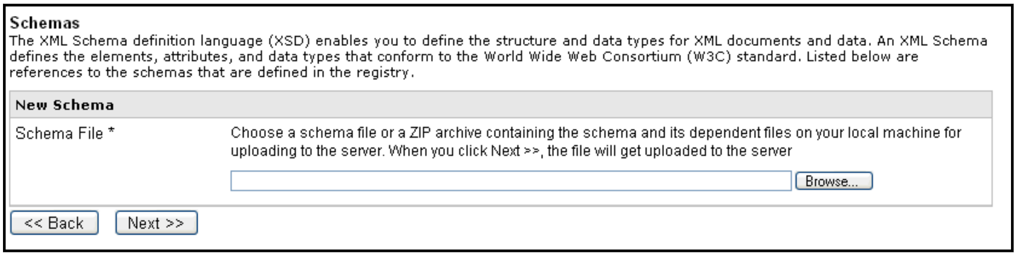
Schemas

□ Filter By Name Where Name Equals

□	Name	Download	References	Description
□	Any			A schema representing a definition for any document in and any document out. This schema has two possible root elements, appropriately named 'request' and 'response'.
□	PFIVP			This schema describes the input document that is passed to the PFIVP sample process, delivered with iWay Designer.
□	PFIVPResponse			This schema describes the document that is returned from the PFIVP sample process, delivered with iWay Designer.
□	pictures.EncodedAsXML			This schema describes an encoded picture. The root element is defined as <picture> and its content is a base64 encoded string that represents the bytes of the picture.
□	pictures.SqlInsertPicture			The schema defines a sql statment that is used to insert a picture into the pictures table using the RDBMS adapter.
□	pictures.SqlQueryPicture			The schema defines the result of issuing a sql select statement that is used to retrieve the images in the pictures table using the RDBMS adapter.

2. Click *Add*.

   The New Schema pane opens.



Schemas

The XML Schema definition language (XSD) enables you to define the structure and data types for XML documents and data. An XML Schema defines the elements, attributes, and data types that conform to the World Wide Web Consortium (W3C) standard. Listed below are references to the schemas that are defined in the registry.

New Schema

Schema File * Choose a schema file or a ZIP archive containing the schema and its dependent files on your local machine for uploading to the server. When you click Next >>, the file will get uploaded to the server

3. Type the path to the XML Schema Definition (XSD) file or .ZIP archive on your file system.

   You can also click *Browse* to find its location.

   Note: If you are uploading a .ZIP file that contains the schema, a manifest file must also be included in the archive. For more information on the format of the manifest file, see *How to Format the Manifest File* on page 283.

4. Click *Next*.

   The Name and Description pane opens.

5. Provide a name and, optionally, a description, for the schema, and click *Finish*.

The schema is uploaded to the server and is added to the list in the Schemas pane.

After they are defined, schemas can be added to the start and end of a process flow. In addition, embedded schemas can be uploaded as one unit in the form of a zip file. However, it requires a manifest.

Procedure: How to Format the Manifest File

To format the manifest file:

1. Create a file called *Manifest.mf* located in the *meta-inf* subfolder.
2. Add a reference to the root schema of your archive on the very first line using the following format:

   ```
   IWAY-File: DIRECTORY/FILENAME.XSD
   ```

   where:

   `DIRECTORY`

   Is the path to the schema file.

   `FILENAME`

   Is the name of the schema file.

   Note: This reference must be typed in uppercase.
3. Make sure that the manifest does not contain any comments or blank lines above the root schema reference line.
4. Add the *meta-inf/Manifest.mf* file to your .ZIP archive.

Adding a Service

Services are executable Java procedures that are used to handle the business logic of a message.

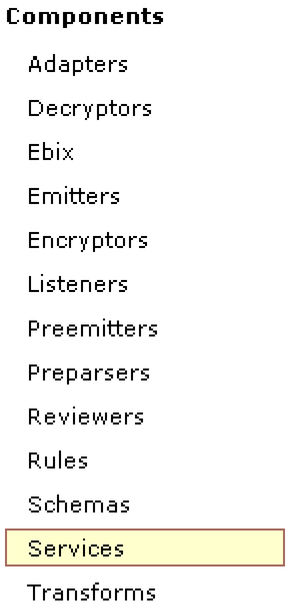
A service is the business layer that incorporates the logic for encapsulating the business process which interacts with other distributed component services to provide transactions for business "state" information. This business layer incorporates the application business logic. In an iWay Service Manager environment, business logic consists of one or more services acting on an input document. Services can be stacked, or multiple services can be executed in parallel. It is recommended that services be added using iIT Designer.

Note: It is strongly recommended to use Process Flows and not stand-alone services.

Services are written in standard Java language and can make use of any available Java libraries or services. iWay Software supplies several predefined services that you can use as part of your business logic. For more information on the specific properties of iSM pre-defined services, see the *iWay Service Manager Component and Functional Language Reference Guide*.

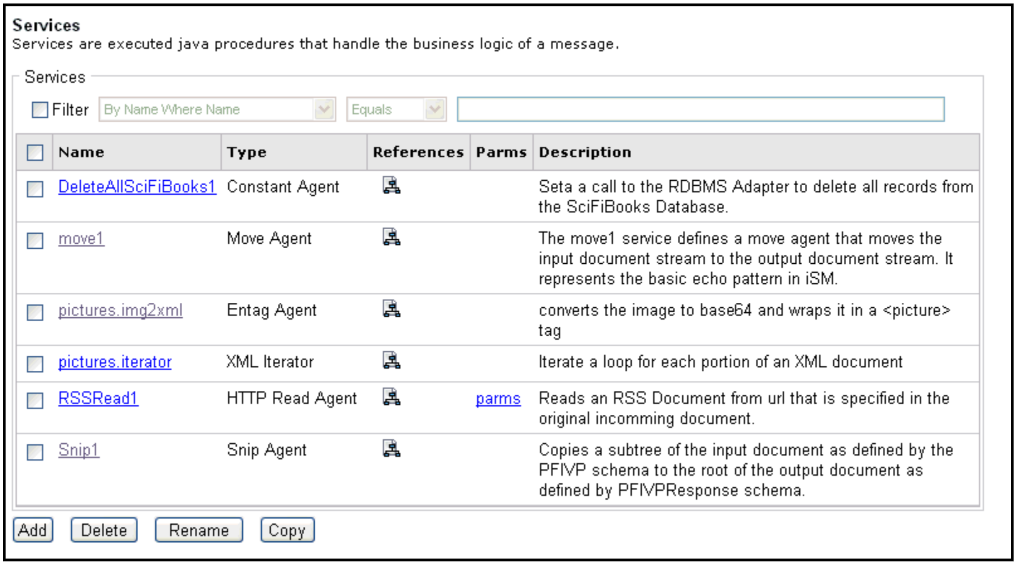
Procedure: How to Configure a Service

To configure a service using the iWay Service Manager Administration Console:



1. In the left console pane of the Registry menu, select *Services*.

The Services pane opens.



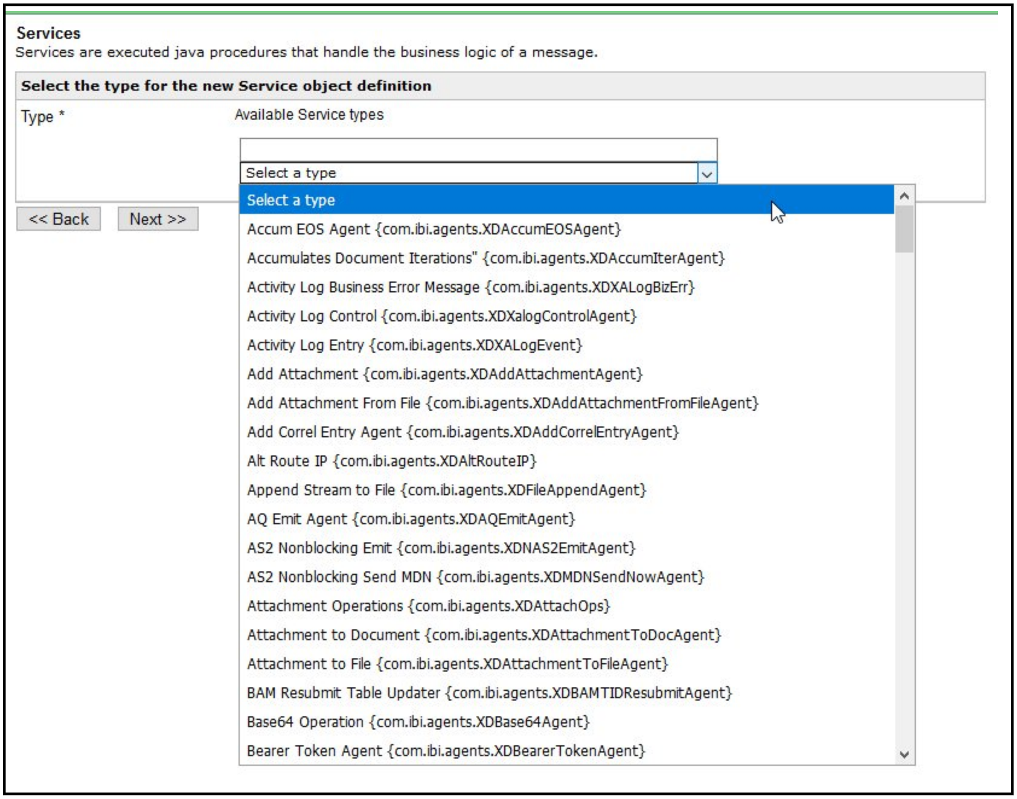
Services

Services are executed java procedures that handle the business logic of a message.

□	Name	Type	References	Parms	Description
□	DeleteAllSciFiBooks1	Constant Agent			Seta a call to the RDBMS Adapter to delete all records from the SciFiBooks Database.
□	move1	Move Agent			The move1 service defines a move agent that moves the input document stream to the output document stream. It represents the basic echo pattern in iSM.
□	pictures.img2xml	Entag Agent			converts the image to base64 and wraps it in a <picture> tag
□	pictures.iterator	XML Iterator			Iterate a loop for each portion of an XML document
□	RSSRead1	HTTP Read Agent		parms	Reads an RSS Document from url that is specified in the original incomming document.
□	Snip1	Snip Agent			Copies a subtree of the input document as defined by the PFIVP schema to the root of the output document as defined by PFIVPResponse schema.

2. Click *Add*.

The Service Type pane opens.



Services

Services are executed java procedures that handle the business logic of a message.

Select the type for the new Service object definition

Type * Available Service types

Select a type

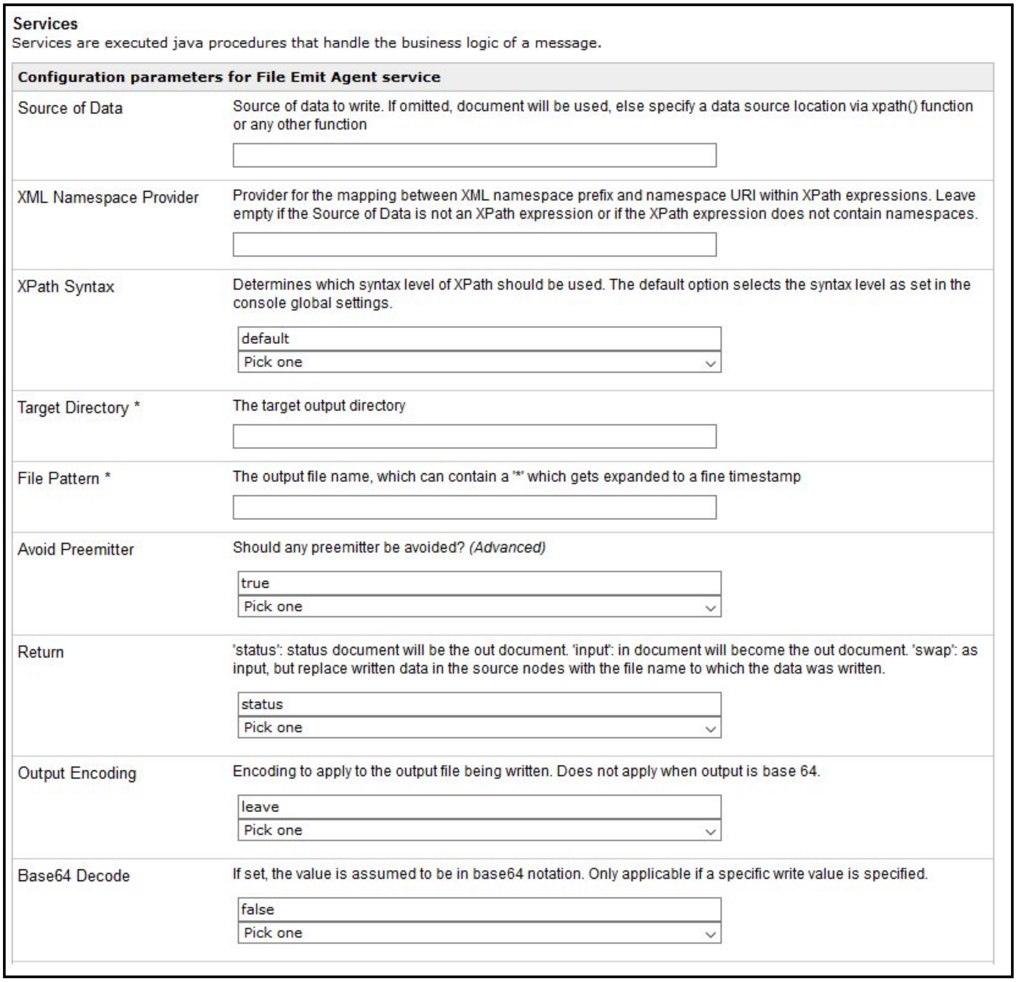
- Select a type
- Accum EOS Agent {com.ibi.agents.XDAccumEOSAgent}
- Accumulates Document Iterations" {com.ibi.agents.XDAccumIterAgent}
- Activity Log Business Error Message {com.ibi.agents.XDXALogBizErr}
- Activity Log Control {com.ibi.agents.XDXalogControlAgent}
- Activity Log Entry {com.ibi.agents.XDXALogEvent}
- Add Attachment {com.ibi.agents.XDAddAttachmentAgent}
- Add Attachment From File {com.ibi.agents.XDAddAttachmentFromFileAgent}
- Add Correl Entry Agent {com.ibi.agents.XDAddCorrelEntryAgent}
- Alt Route IP {com.ibi.agents.XDAltRouteIP}
- Append Stream to File {com.ibi.agents.XDFileAppendAgent}
- AQ Emit Agent {com.ibi.agents.XDAQEmitAgent}
- AS2 Nonblocking Emit {com.ibi.agents.XDNAS2EmitAgent}
- AS2 Nonblocking Send MDN {com.ibi.agents.XDMDNSendNowAgent}
- Attachment Operations {com.ibi.agents.XDAttachOps}
- Attachment to Document {com.ibi.agents.XDAttachmentToDocAgent}
- Attachment to File {com.ibi.agents.XDAttachmentToFileAgent}
- BAM Resubmit Table Updater {com.ibi.agents.XDBAMTIDResubmitAgent}
- Base64 Operation {com.ibi.agents.XDBase64Agent}
- Bearer Token Agent {com.ibi.agents.XDBearerTokenAgent}

3. Select a type of service from the list, for example, File Emit Agent (com.ibi.agents.XDFileEmitAgent).

   You can also type the class name of a service that is packaged in a .JAR file, which is available in the Java class path.

4. Click *Next*.

The configuration parameters pane for the File Emit Agent service opens.



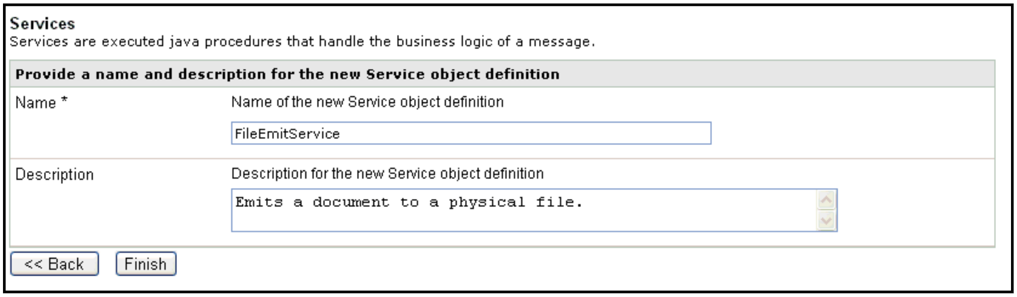
Services

Services are executed java procedures that handle the business logic of a message.

Configuration parameters for File Emit Agent service		
Source of Data	Source of data to write. If omitted, document will be used, else specify a data source location via xpath() function or any other function	
XML Namespace Provider	Provider for the mapping between XML namespace prefix and namespace URI within XPath expressions. Leave empty if the Source of Data is not an XPath expression or if the XPath expression does not contain namespaces.	
XPath Syntax	Determines which syntax level of XPath should be used. The default option selects the syntax level as set in the console global settings.	default Pick one
Target Directory *	The target output directory	
File Pattern *	The output file name, which can contain a '*' which gets expanded to a fine timestamp	
Avoid Preemitter	Should any preemitter be avoided? *(Advanced)*	true Pick one
Return	'status': status document will be the out document. 'input': in document will become the out document. 'swap': as input, but replace written data in the source nodes with the file name to which the data was written.	status Pick one
Output Encoding	Encoding to apply to the output file being written. Does not apply when output is base 64.	leave Pick one
Base64 Decode	If set, the value is assumed to be in base64 notation. Only applicable if a specific write value is specified.	false Pick one

5. Provide the required configuration parameters for the service, and click *Next*.

The Name and Description pane opens.



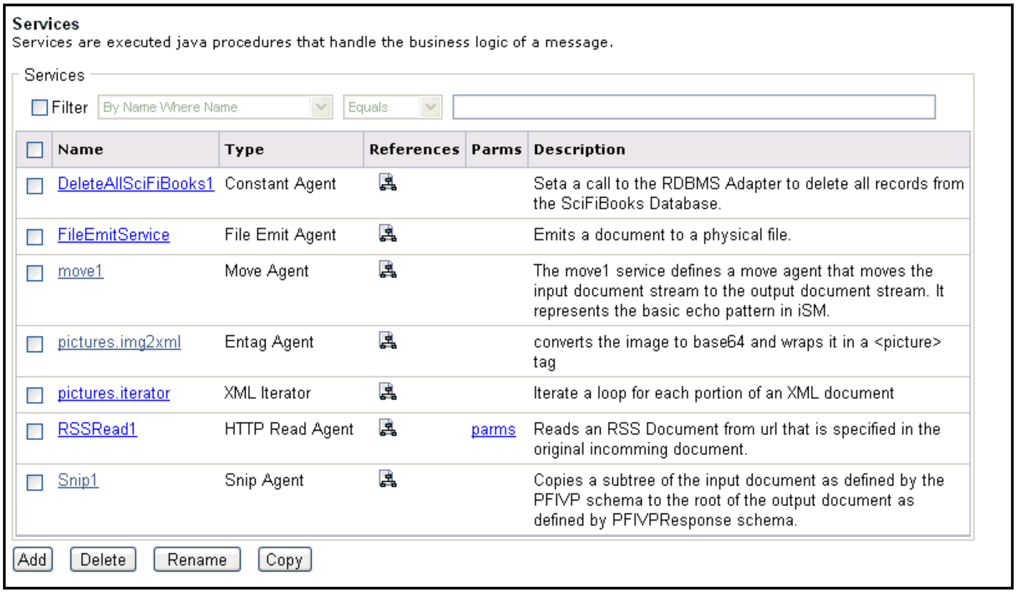
Services
Services are executed java procedures that handle the business logic of a message.

Provide a name and description for the new Service object definition

Name *	Name of the new Service object definition FileEmitService
Description	Description for the new Service object definition Emits a document to a physical file.

6. Provide a name and, optionally, a description, for the service, and click *Finish*.

   The service is added to the list in the Services pane.



Services
Services are executed java procedures that handle the business logic of a message.

Services

□ Filter By Name Where Name Equals

□	Name	Type	References	Parms	Description
□	DeleteAllSciFiBooks1	Constant Agent	🖧		Seta a call to the RDBMS Adapter to delete all records from the SciFiBooks Database.
□	FileEmitService	File Emit Agent	🖧		Emits a document to a physical file.
□	move1	Move Agent	🖧		The move1 service defines a move agent that moves the input document stream to the output document stream. It represents the basic echo pattern in iSM.
□	pictures.img2xml	Entag Agent	🖧		converts the image to base64 and wraps it in a <picture> tag
□	pictures.iterator	XML Iterator	🖧		Iterate a loop for each portion of an XML document
□	RSSRead1	HTTP Read Agent	🖧	parms	Reads an RSS Document from url that is specified in the original incomming document.
□	Snip1	Snip Agent	🖧		Copies a subtree of the input document as defined by the PFIVP schema to the root of the output document as defined by PFIVPResponse schema.

After a service is added to iWay Service Manager, you can assign it to a process that is used to configure a route. For more information, see *Configuring Channels* on page 305.

Multiple Services

If your requirements include using multiple services for complex business logic, you must create a process flow with multiple services using iIT Designer. For more information on creating process flows, see the *iWay Integration Tools Designer User's Guide*.

Conditional Routes

In iWay Service Manager, a route has the responsibility for handling one aspect of document processing. When multiple routes are defined, configuring conditional settings at the route level allows you to control which route is executed. For more information, see *Configuring Channels* on page 305.

Configuring Acknowledgment Services

Acknowledgment services return an acknowledgment document to a client indicating that a request or input document was received. You can use acknowledgment services with adapters such as the iWay Adapter for EDI X12. For more information, see the *iWay Integration Solution for EDI User's Guide*.

The acknowledgment document also indicates whether the input is valid and is being processed or if the input failed validation. The actual processing of the document is performed by other configured services, for example, TRANSFORM and others.

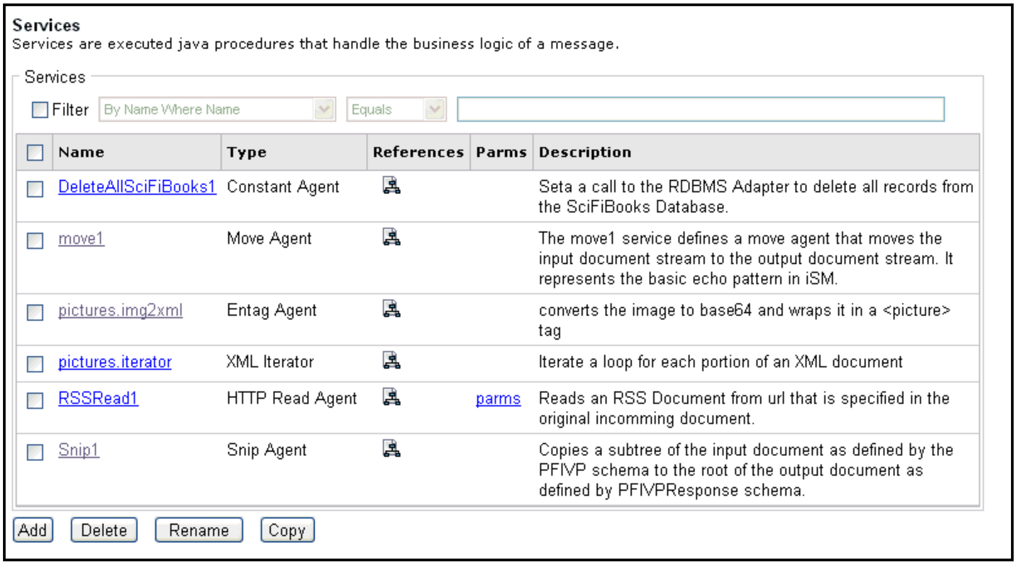
During processing of acknowledgment services, the special register *ackresponse* is set to '1'. This simplifies condition testing for the associated emitter components so that some are used for sending the acknowledgment and some are used for other purposes. The register is set regardless of whether an acknowledgment or a NAK is being emitted by the service.

Procedure: How to Define an Acknowledgment Service

To define an acknowledgment service:

1. In the left console pane of the Registry menu, select *Services*.

The Services pane opens.



Services
Services are executed java procedures that handle the business logic of a message.

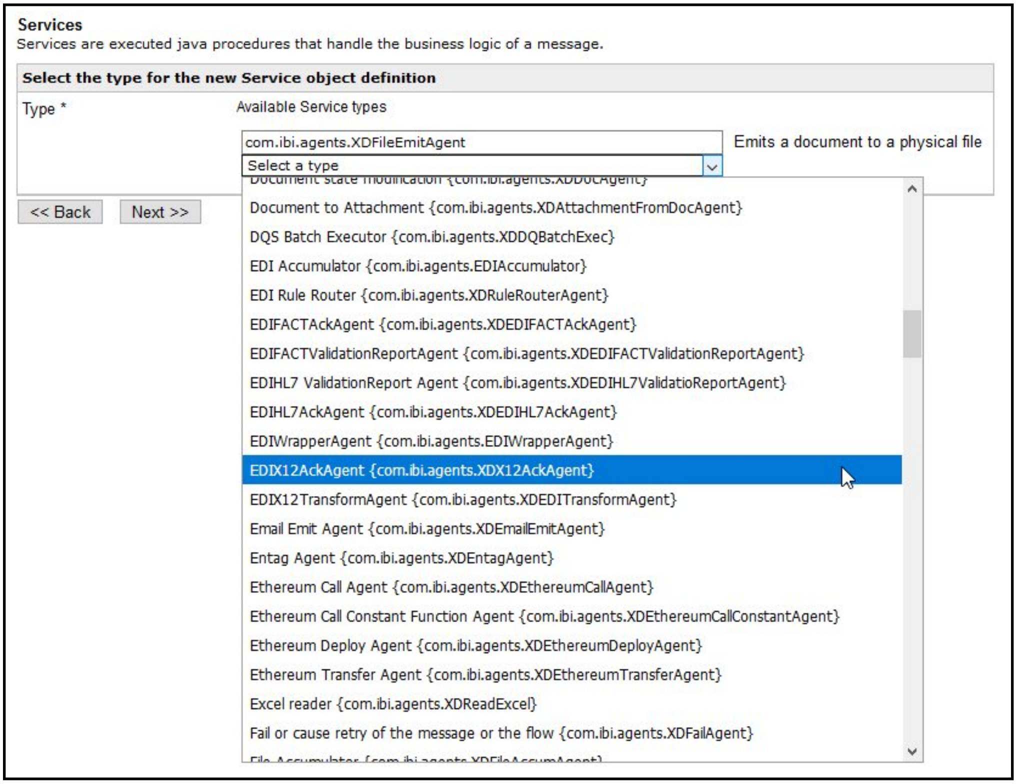
Services

□ Filter

□	Name	Type	References	Parms	Description
□	DeleteAllSciFiBooks1	Constant Agent	🖧		Seta a call to the RDBMS Adapter to delete all records from the SciFiBooks Database.
□	move1	Move Agent	🖧		The move1 service defines a move agent that moves the input document stream to the output document stream. It represents the basic echo pattern in iSM.
□	pictures.img2xml	Entag Agent	🖧		converts the image to base64 and wraps it in a <picture> tag
□	pictures.iterator	XML Iterator	🖧		Iterate a loop for each portion of an XML document
□	RSSRead1	HTTP Read Agent	🖧	parms	Reads an RSS Document from url that is specified in the original incomming document.
□	Snip1	Snip Agent	🖧		Copies a subtree of the input document as defined by the PFIVP schema to the root of the output document as defined by PFIVPResponse schema.

2. Click *Add*.

The Service Type pane opens.



Services

Services are executed java procedures that handle the business logic of a message.

Select the type for the new Service object definition

Type *

Available Service types

com.ibi.agents.XDFileEmitAgent

Emits a document to a physical file

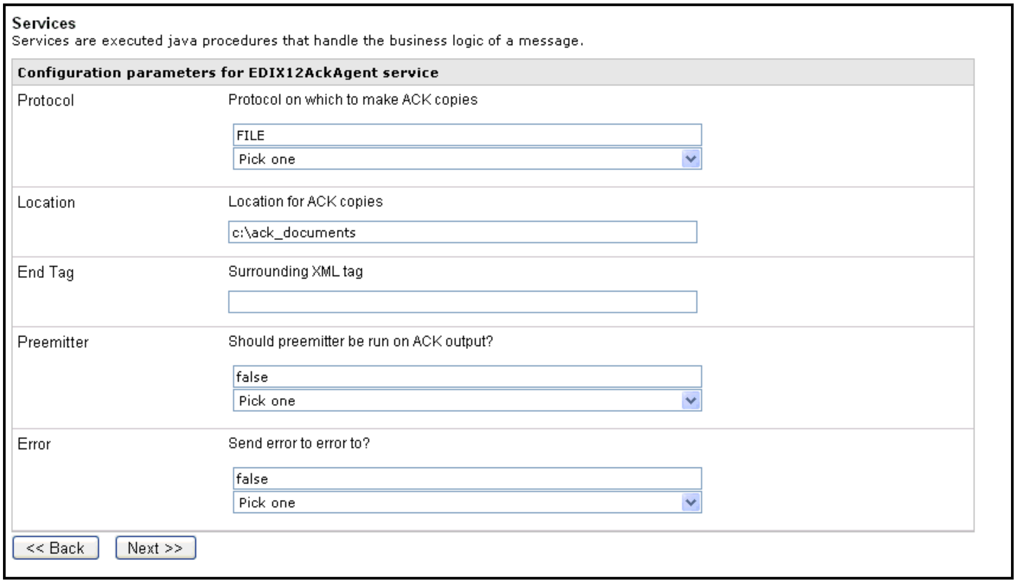
Select a type

Document state modification {com.ibi.agents.XDDocAgent}
Document to Attachment {com.ibi.agents.XDAttachmentFromDocAgent}
DQS Batch Executor {com.ibi.agents.XDDQBatchExec}
EDI Accumulator {com.ibi.agents.EDIAccumulator}
EDI Rule Router {com.ibi.agents.XDRuleRouterAgent}
EDIFACTAckAgent {com.ibi.agents.XDEDIFACTAckAgent}
EDIFACTValidationReportAgent {com.ibi.agents.XDEDIFACTValidationReportAgent}
EDIHL7 ValidationReport Agent {com.ibi.agents.XDEDIHL7ValidatioReportAgent}
EDIHL7AckAgent {com.ibi.agents.XDEDIHL7AckAgent}
EDIWrapperAgent {com.ibi.agents.EDIWrapperAgent}
EDIX12AckAgent {com.ibi.agents.XDX12AckAgent}
EDIX12TransformAgent {com.ibi.agents.XDEDITransformAgent}
Email Emit Agent {com.ibi.agents.XDEmailEmitAgent}
Entag Agent {com.ibi.agents.XDEntagAgent}
Ethereum Call Agent {com.ibi.agents.XDEthereumCallAgent}
Ethereum Call Constant Function Agent {com.ibi.agents.XDEthereumCallConstantAgent}
Ethereum Deploy Agent {com.ibi.agents.XDEthereumDeployAgent}
Ethereum Transfer Agent {com.ibi.agents.XDEthereumTransferAgent}
Excel reader {com.ibi.agents.XDReadExcel}
Fail or cause retry of the message or the flow {com.ibi.agents.XDFailAgent}
File Accumulator {com.ibi.agents.XDFileAccumAgent}

3. Select *EDIX12AckAgent (com.ibi.agents.XDX12AckAgent)* from the drop-down list.

4. Click *Next*.

The configuration parameters pane for the EDIX12AckAgent service opens.



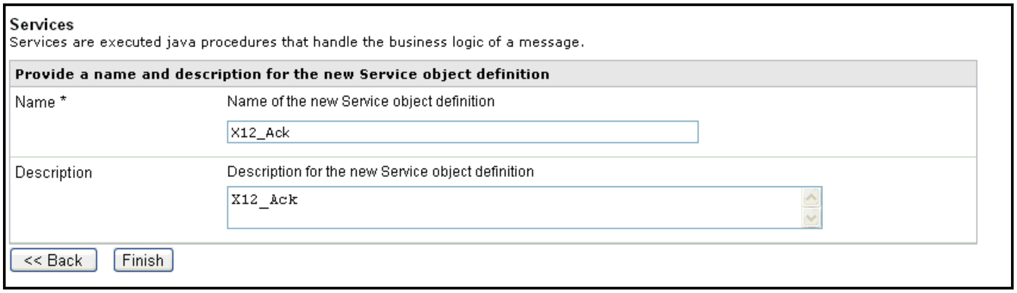
Services
Services are executed java procedures that handle the business logic of a message.

Configuration parameters for EDIX12AckAgent service

Protocol	Protocol on which to make ACK copies	FILE Pick one
Location	Location for ACK copies	c:\ack_documents
End Tag	Surrounding XML tag	
Preemitter	Should preemitter be run on ACK output?	false Pick one
Error	Send error to error to?	false Pick one

5. Select the default parameters for this service, and click *Next*.

The Name and Description pane opens.



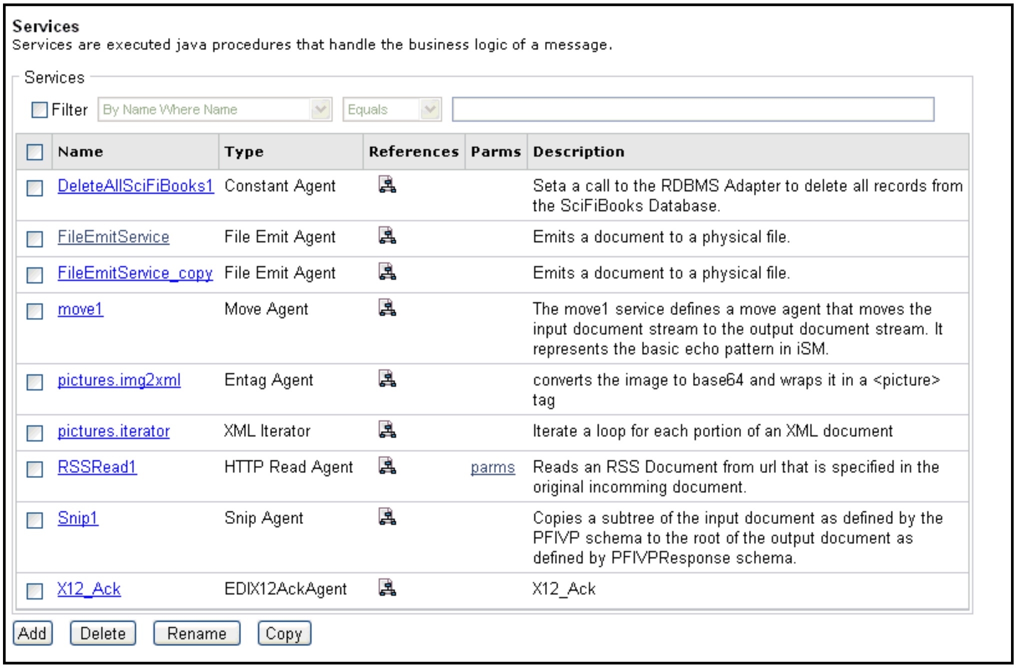
Services
Services are executed java procedures that handle the business logic of a message.

Provide a name and description for the new Service object definition

Name *	Name of the new Service object definition	X12_Ack
Description	Description for the new Service object definition	X12_Ack

6. Provide a name, for example, X12_Ack, and optionally, a description, for this service, and click *Finish*.

The service is added to the list in the Services pane.



Services

Services are executed java procedures that handle the business logic of a message.

□	Name	Type	References	Parms	Description
□	DeleteAllSciFiBooks1	Constant Agent	🖧		Seta a call to the RDBMS Adapter to delete all records from the SciFiBooks Database.
□	FileEmitService	File Emit Agent	🖧		Emits a document to a physical file.
□	FileEmitService_copy	File Emit Agent	🖧		Emits a document to a physical file.
□	move1	Move Agent	🖧		The move1 service defines a move agent that moves the input document stream to the output document stream. It represents the basic echo pattern in iSM.
□	pictures.img2xml	Entag Agent	🖧		converts the image to base64 and wraps it in a <picture> tag
□	pictures.iterator	XML Iterator	🖧		Iterate a loop for each portion of an XML document
□	RSSRead1	HTTP Read Agent	🖧	parms	Reads an RSS Document from url that is specified in the original incomming document.
□	Snip1	Snip Agent	🖧		Copies a subtree of the input document as defined by the PFIVP schema to the root of the output document as defined by PFIVPResponse schema.
□	X12_Ack	EDIX12AckAgent	🖧		X12_Ack

After the EDIX12AckAgent service is defined, you can create a new process and assign this service to that process. For more information, see the *iWay Integration Solution for EDI User's Guide*.

Viewing a Defined Transform

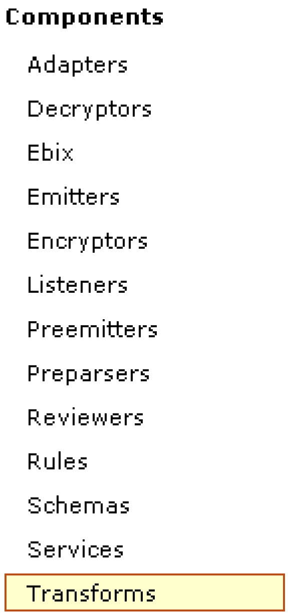
Transformation definition files contain sets of rules, interpreted and executed by a transformation engine. Transformation is the process by which data is transformed from one structure/format to another.

Note: Any transformation projects that are published to the registry using iWay Transformer are also made available in the Transforms pane. It is recommended to use iWay Transformer when you are publishing any transformation project.

For information on creating transformation projects and publishing them, see the *iWay Transformer User's Guide*.

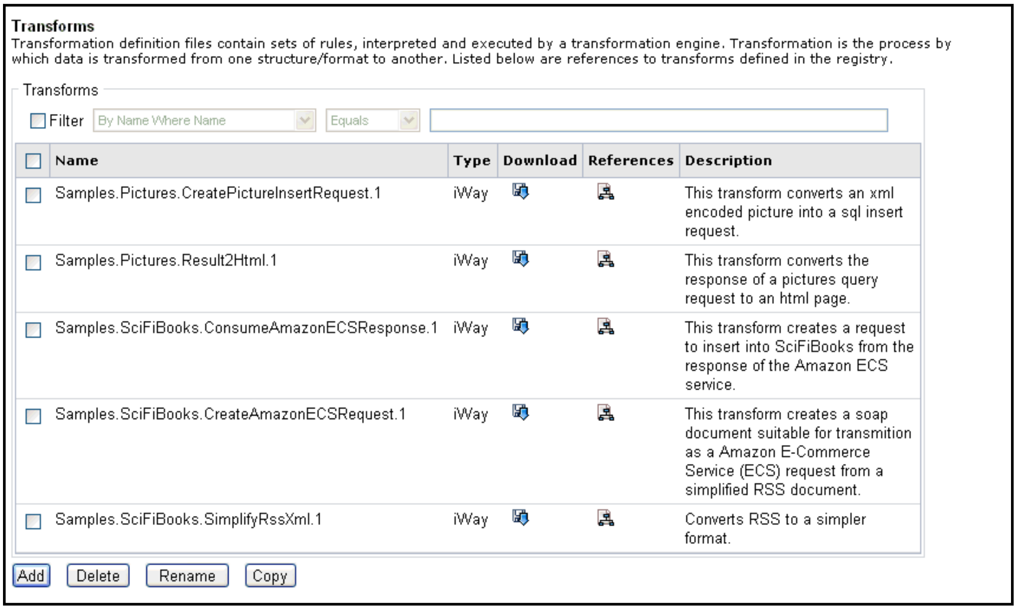
Procedure: How to View a Transform

To view a transform:



1. In the left console pane of the Registry menu, select *Transforms*.

The Transforms pane opens.



Transforms

Transformation definition files contain sets of rules, interpreted and executed by a transformation engine. Transformation is the process by which data is transformed from one structure/format to another. Listed below are references to transforms defined in the registry.

Transforms

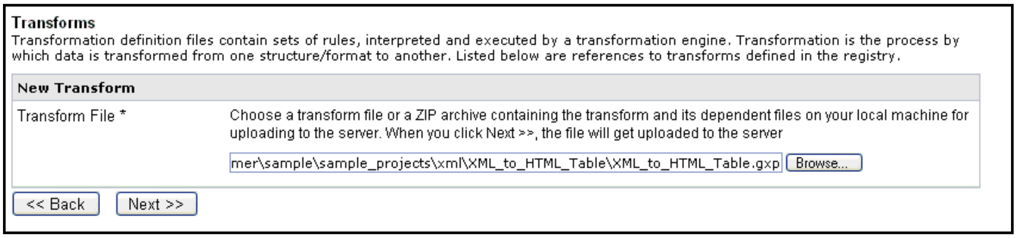
□ Filter [By Name Where Name ▾] [Equals ▾] [ ]

□	Name	Type	Download	References	Description
□	Samples.Pictures.CreatePictureInsertRequest.1	iWay	⬇	⧉	This transform converts an xml encoded picture into a sql insert request.
□	Samples.Pictures.Result2Html.1	iWay	⬇	⧉	This transform converts the response of a pictures query request to an html page.
□	Samples.SciFiBooks.ConsumeAmazonECSResponse.1	iWay	⬇	⧉	This transform creates a request to insert into SciFiBooks from the response of the Amazon ECS service.
□	Samples.SciFiBooks.CreateAmazonECSRequest.1	iWay	⬇	⧉	This transform creates a soap document suitable for transmition as a Amazon E-Commerce Service (ECS) request from a simplified RSS document.
□	Samples.SciFiBooks.SimplifyRssXml.1	iWay	⬇	⧉	Converts RSS to a simpler format.

The table that is provided lists all the transforms that are currently available in the registry. You can also publish a transform into the registry using iWay Transformer. For more information, see the *iWay Transformer User's Guide*.

2. Click *Add*.

   The New Transform pane opens.



Transforms

Transformation definition files contain sets of rules, interpreted and executed by a transformation engine. Transformation is the process by which data is transformed from one structure/format to another. Listed below are references to transforms defined in the registry.

New Transform

Transform File * — Choose a transform file or a ZIP archive containing the transform and its dependent files on your local machine for uploading to the server. When you click Next >>, the file will get uploaded to the server

mer\sample\sample_projects\xml\XML_to_HTML_Table\XML_to_HTML_Table.gxp

3. Type the path to the transform project file (.GXP) on your file system or click *Browse* to find its location.

4. Click *Next*.

   The Name and Description pane opens.

5. Provide a name and, optionally, a description, for the transform, and click *Finish*.

   The transform is uploaded to the server and is added to the list in the Transforms pane.

Defining a Process

Processes are stateless, lightweight, short-lived microflows that are executed by iWay Service Manager on messages/documents as they pass through the system. Processes that are published using iIT Designer are available in the registry and can be bound to channels as routes.

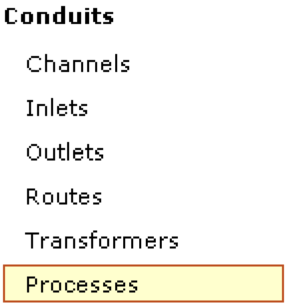
Processes can be published directly from iIT Designer to the registry. For information on creating and publishing a process, see the *iWay Integration Tools Designer User's Guide*.

As an alternative, the following section shows how to construct processes using one or more of the services that have already been defined.

Note: As a best practice, it is recommended to use iIT Designer to define and publish a process.

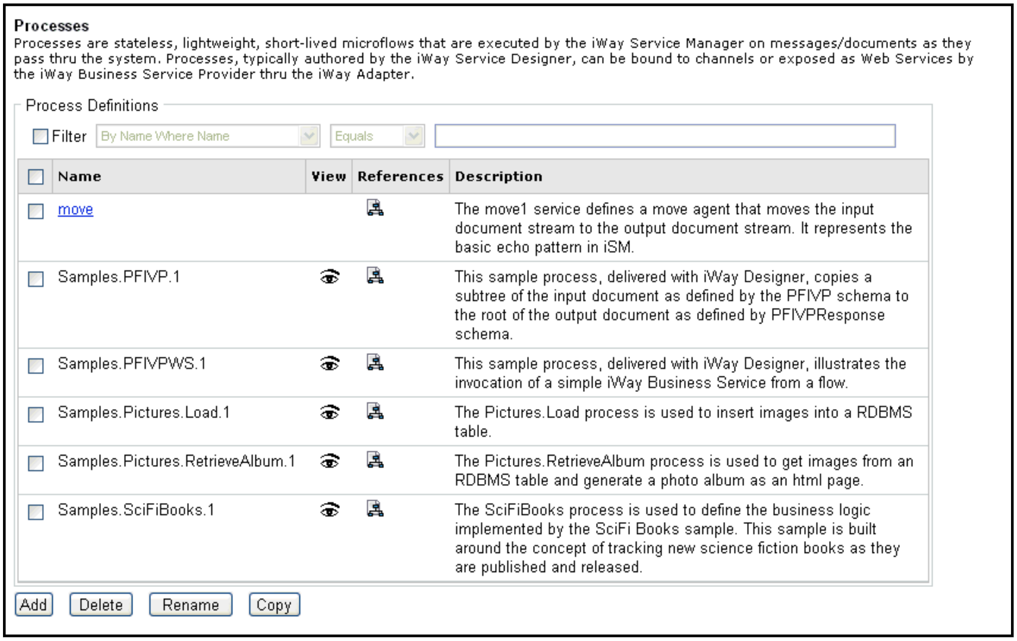
Procedure: How to Define a Process

To define a process using the iWay Service Manager Administration Console:



1. In the left console pane of the Registry menu, select *Processes*.

The Processes pane opens.



Processes

Processes are stateless, lightweight, short-lived microflows that are executed by the iWay Service Manager on messages/documents as they pass thru the system. Processes, typically authored by the iWay Service Designer, can be bound to channels or exposed as Web Services by the iWay Business Service Provider thru the iWay Adapter.

Process Definitions

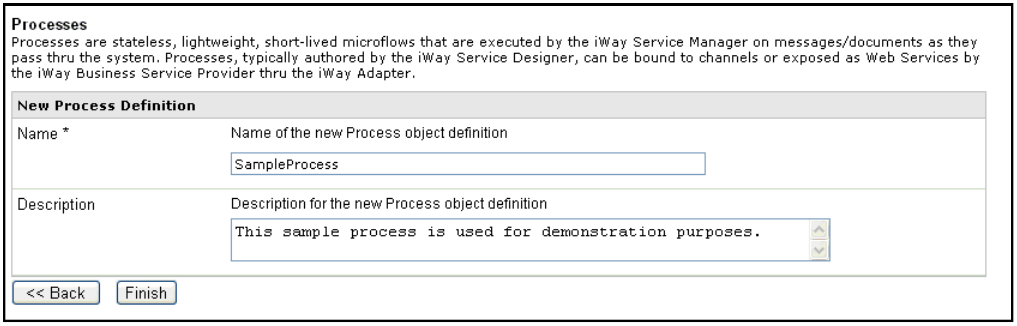
☐ Filter | By Name Where Name | Equals |

☐	Name	View	References	Description
☐	move			The move1 service defines a move agent that moves the input document stream to the output document stream. It represents the basic echo pattern in iSM.
☐	Samples.PFIVP.1			This sample process, delivered with iWay Designer, copies a subtree of the input document as defined by the PFIVP schema to the root of the output document as defined by PFIVPResponse schema.
☐	Samples.PFIVPWS.1			This sample process, delivered with iWay Designer, illustrates the invocation of a simple iWay Business Service from a flow.
☐	Samples.Pictures.Load.1			The Pictures.Load process is used to insert images into a RDBMS table.
☐	Samples.Pictures.RetrieveAlbum.1			The Pictures.RetrieveAlbum process is used to get images from an RDBMS table and generate a photo album as an html page.
☐	Samples.SciFiBooks.1			The SciFiBooks process is used to define the business logic implemented by the SciFi Books sample. This sample is built around the concept of tracking new science fiction books as they are published and released.

The table that is provided lists all defined processes that are currently available. If you click the document schematic icon in the References column for a specific process, you will see which components are referencing that process. If you click the eye icon in the View column for a specific process, you will see a visual depiction of that process.

2. Click *Add*.

   The New Process Definition pane opens.



Processes

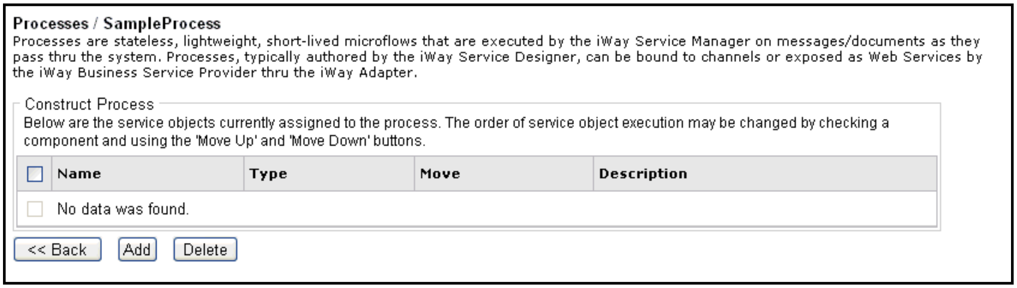
Processes are stateless, lightweight, short-lived microflows that are executed by the iWay Service Manager on messages/documents as they pass thru the system. Processes, typically authored by the iWay Service Designer, can be bound to channels or exposed as Web Services by the iWay Business Service Provider thru the iWay Adapter.

New Process Definition

Name *	Name of the new Process object definition SampleProcess
Description	Description for the new Process object definition This sample process is used for demonstration purposes.

3. Provide a name and, optionally, a description, for the process, and click *Finish*.

The Construct Process pane opens.



Processes / SampleProcess

Processes are stateless, lightweight, short-lived microflows that are executed by the iWay Service Manager on messages/documents as they pass thru the system. Processes, typically authored by the iWay Service Designer, can be bound to channels or exposed as Web Services by the iWay Business Service Provider thru the iWay Adapter.

Construct Process

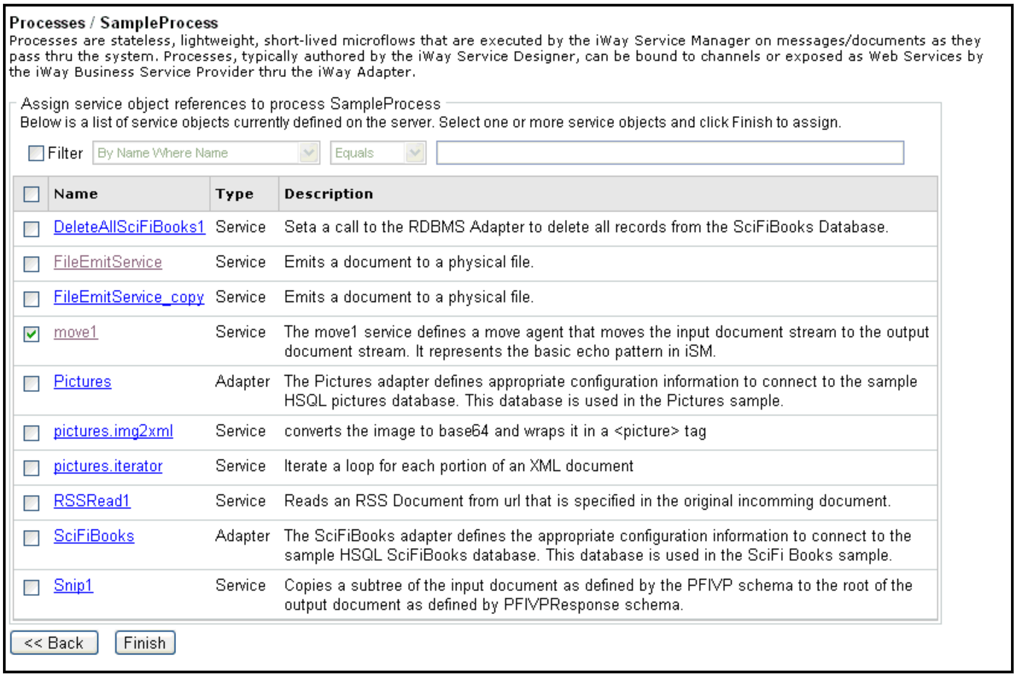
Below are the service objects currently assigned to the process. The order of service object execution may be changed by checking a component and using the 'Move Up' and 'Move Down' buttons.

□	Name	Type	Move	Description
□	No data was found.			

The table that is provided lists the service objects that are currently assigned to the process. When multiple service objects are added, you can modify their processing order by using the *Move Up* and *Move Down* buttons that become available in the Move column.

4. Click *Add*.

   The Assign service object references pane for the process you are currently defining opens.



Processes / SampleProcess

Processes are stateless, lightweight, short-lived microflows that are executed by the iWay Service Manager on messages/documents as they pass thru the system. Processes, typically authored by the iWay Service Designer, can be bound to channels or exposed as Web Services by the iWay Business Service Provider thru the iWay Adapter.

Assign service object references to process SampleProcess

Below is a list of service objects currently defined on the server. Select one or more service objects and click Finish to assign.

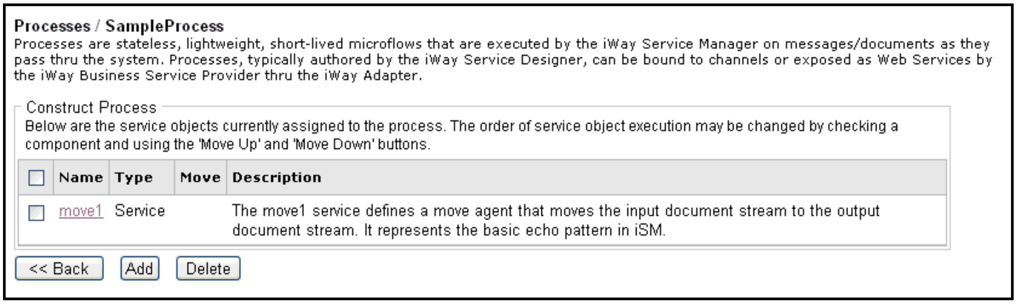
□ Filter By Name Where Name Equals

□	Name	Type	Description
□	DeleteAllSciFiBooks1	Service	Seta a call to the RDBMS Adapter to delete all records from the SciFiBooks Database.
□	FileEmitService	Service	Emits a document to a physical file.
□	FileEmitService_copy	Service	Emits a document to a physical file.
☑	move1	Service	The move1 service defines a move agent that moves the input document stream to the output document stream. It represents the basic echo pattern in iSM.
□	Pictures	Adapter	The Pictures adapter defines appropriate configuration information to connect to the sample HSQL pictures database. This database is used in the Pictures sample.
□	pictures.img2xml	Service	converts the image to base64 and wraps it in a <picture> tag
□	pictures.iterator	Service	Iterate a loop for each portion of an XML document
□	RSSRead1	Service	Reads an RSS Document from url that is specified in the original incomming document.
□	SciFiBooks	Adapter	The SciFiBooks adapter defines the appropriate configuration information to connect to the sample HSQL SciFiBooks database. This database is used in the SciFi Books sample.
□	Snip1	Service	Copies a subtree of the input document as defined by the PFIVP schema to the root of the output document as defined by PFIVPResponse schema.

The table that is provided lists the service objects that are currently defined in the registry.

5. Select one or more service objects, and click *Finish*.

You are returned to the Construct Process pane.



Processes / SampleProcess

Processes are stateless, lightweight, short-lived microflows that are executed by the iWay Service Manager on messages/documents as they pass thru the system. Processes, typically authored by the iWay Service Designer, can be bound to channels or exposed as Web Services by the iWay Business Service Provider thru the iWay Adapter.

Construct Process

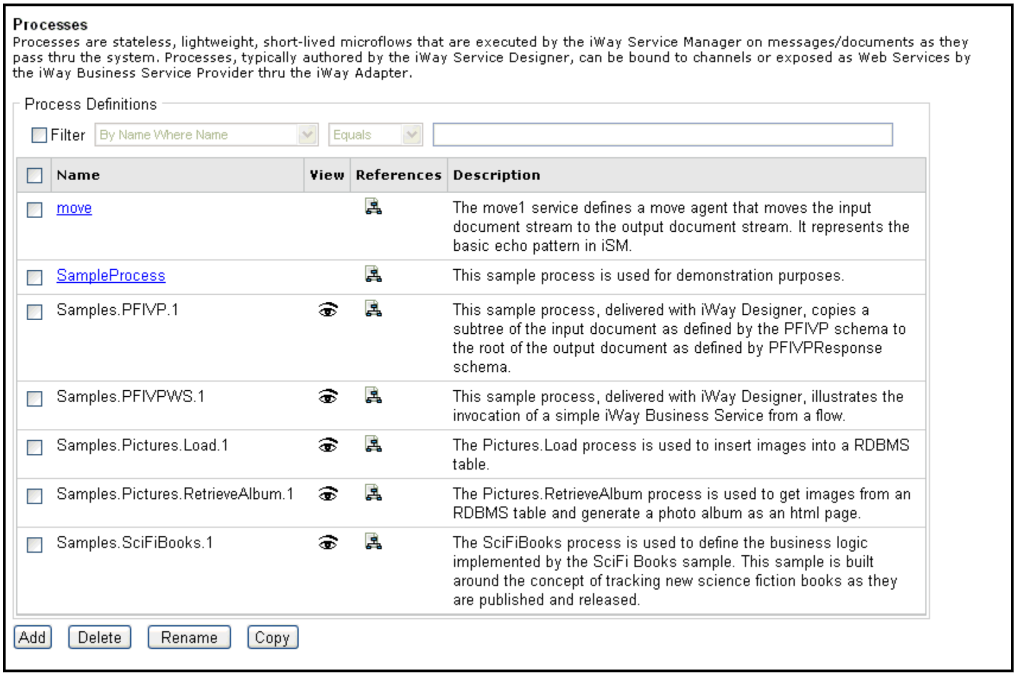
Below are the service objects currently assigned to the process. The order of service object execution may be changed by checking a component and using the 'Move Up' and 'Move Down' buttons.

□	Name	Type	Move	Description
□	move1	Service		The move1 service defines a move agent that moves the input document stream to the output document stream. It represents the basic echo pattern in iSM.

The service object you selected, for example, move1, is added to the table.

6. Return to the main Processes pane.

   The new process you defined is added.



Processes

Processes are stateless, lightweight, short-lived microflows that are executed by the iWay Service Manager on messages/documents as they pass thru the system. Processes, typically authored by the iWay Service Designer, can be bound to channels or exposed as Web Services by the iWay Business Service Provider thru the iWay Adapter.

Process Definitions

□ Filter By Name Where Name Equals

□	Name	View	References	Description
□	move		🖧	The move1 service defines a move agent that moves the input document stream to the output document stream. It represents the basic echo pattern in iSM.
□	SampleProcess		🖧	This sample process is used for demonstration purposes.
□	Samples.PFIVP.1	👁	🖧	This sample process, delivered with iWay Designer, copies a subtree of the input document as defined by the PFIVP schema to the root of the output document as defined by PFIVPResponse schema.
□	Samples.PFIVPWS.1	👁	🖧	This sample process, delivered with iWay Designer, illustrates the invocation of a simple iWay Business Service from a flow.
□	Samples.Pictures.Load.1	👁	🖧	The Pictures.Load process is used to insert images into a RDBMS table.
□	Samples.Pictures.RetrieveAlbum.1	👁	🖧	The Pictures.RetrieveAlbum process is used to get images from an RDBMS table and generate a photo album as an html page.
□	Samples.SciFiBooks.1	👁	🖧	The SciFiBooks process is used to define the business logic implemented by the SciFi Books sample. This sample is built around the concept of tracking new science fiction books as they are published and released.

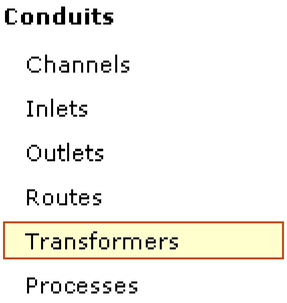
After a process is defined to iWay Service Manager, you can bind it to a channel as a route. For more information, see *Configuring Channels* on page 305.

Defining a Transformer

Transformers are sequences of exits that manipulate a message as it passes through a channel. A transformer can be applied before and after a process and can be included in a **route**.

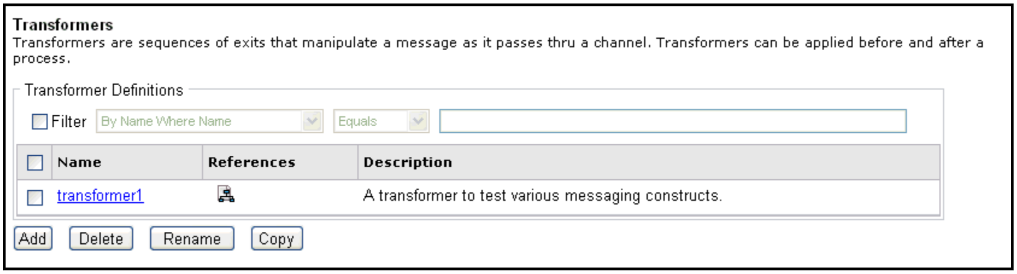
Procedure: How to Define a Transformer

To define a transformer using the iWay Service Manager Administration Console:



1. In the left console pane of the Registry menu, select *Transformers*.

   The Transformers pane opens.



Transformers

Transformers are sequences of exits that manipulate a message as it passes thru a channel. Transformers can be applied before and after a process.

Transformer Definitions

☐ Filter By Name Where Name | Equals

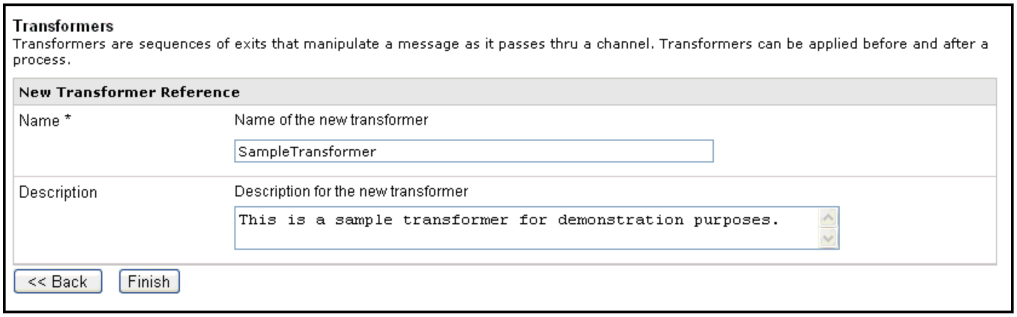
☐	Name	References	Description
☐	transformer1		A transformer to test various messaging constructs.

The table that is provided lists all defined transformers that are currently available. If you click the document schematic icon in the References column for a specific transformer, you will see which components are referencing that transformer.

Note: In most cases, users simply include the transformers within available process flows.

2. Click *Add*.

The New Transformer Reference pane opens.



Transformers

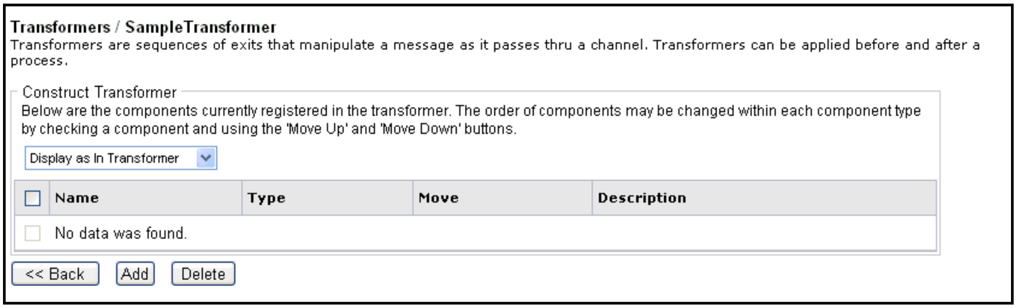
Transformers are sequences of exits that manipulate a message as it passes thru a channel. Transformers can be applied before and after a process.

New Transformer Reference

Name *	Name of the new transformer SampleTransformer
Description	Description for the new transformer This is a sample transformer for demonstration purposes.

3. Provide a name and, optionally, a description, for the transformer, and click *Finish*.

   The Construct Transformer pane opens.



Transformers / SampleTransformer

Transformers are sequences of exits that manipulate a message as it passes thru a channel. Transformers can be applied before and after a process.

Construct Transformer

Below are the components currently registered in the transformer. The order of components may be changed within each component type by checking a component and using the 'Move Up' and 'Move Down' buttons.

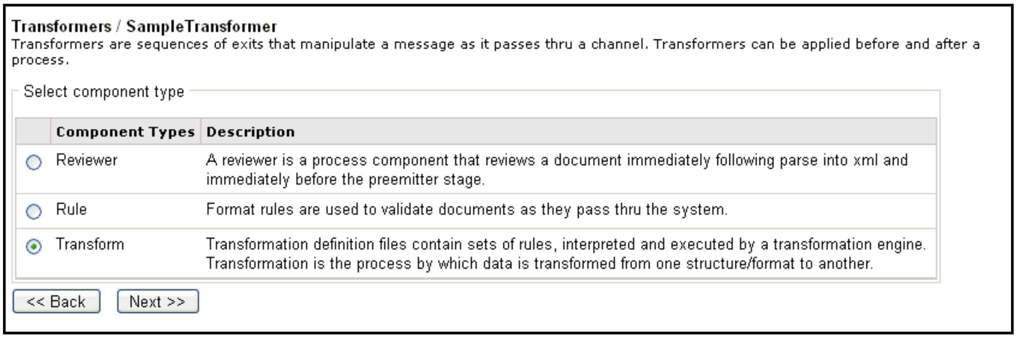
Display as In Transformer

□	Name	Type	Move	Description
□	No data was found.			

The table that is provided lists the components that are currently assigned to the transformer. When multiple components are added, you can modify their processing order by using the *Move Up* and *Move Down* buttons that become available in the Move column.

4. Click *Add*.

   The Select component type pane for the transformer you are currently defining opens.



Transformers / SampleTransformer

Transformers are sequences of exits that manipulate a message as it passes thru a channel. Transformers can be applied before and after a process.

Select component type

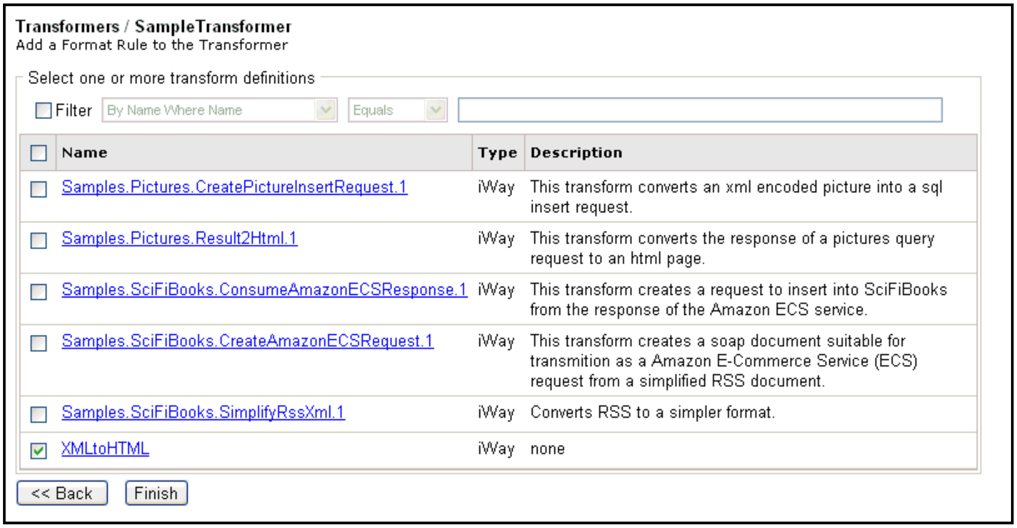
	Component Types	Description
○	Reviewer	A reviewer is a process component that reviews a document immediately following parse into xml and immediately before the preemitter stage.
○	Rule	Format rules are used to validate documents as they pass thru the system.
●	Transform	Transformation definition files contain sets of rules, interpreted and executed by a transformation engine. Transformation is the process by which data is transformed from one structure/format to another.

The table provides a list of supported components that you can assign to a transformer.

- **Reviewer** - Process component that reviews a document immediately following the preparsing stage into XML and immediately before the preemitter stage. For more information, see *Adding a Reviewer* on page 275. It can have an inbound and outbound transformer.
- **Rule** - Format rules used to validate documents as they pass through the system.
- **Transform** - Transformation definition files containing sets of rules, interpreted and executed by a transformation engine. Transformation is the process by which data is transformed from one structure/format to another.

5. Select one of the components, for example, *Transform*, and click *Next*.

   The Transform definitions pane opens.



Transformers / SampleTransformer
Add a Format Rule to the Transformer

Select one or more transform definitions

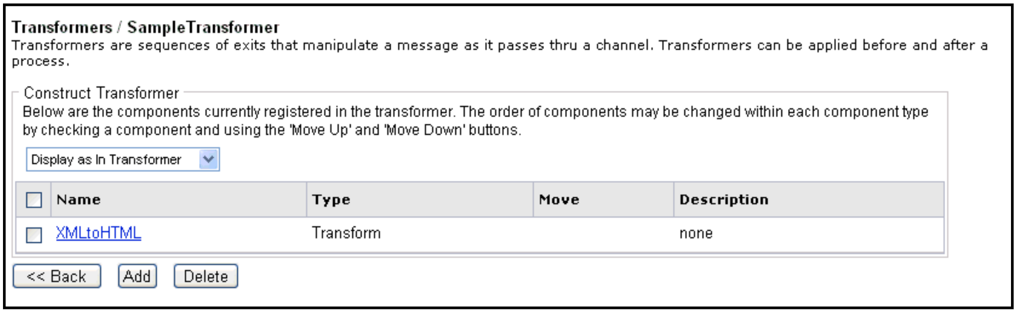
□ Filter By Name Where Name Equals

□	Name	Type	Description
□	Samples.Pictures.CreatePictureInsertRequest.1	iWay	This transform converts an xml encoded picture into a sql insert request.
□	Samples.Pictures.Result2Html.1	iWay	This transform converts the response of a pictures query request to an html page.
□	Samples.SciFiBooks.ConsumeAmazonECSResponse.1	iWay	This transform creates a request to insert into SciFiBooks from the response of the Amazon ECS service.
□	Samples.SciFiBooks.CreateAmazonECSRequest.1	iWay	This transform creates a soap document suitable for transmition as a Amazon E-Commerce Service (ECS) request from a simplified RSS document.
□	Samples.SciFiBooks.SimplifyRssXml.1	iWay	Converts RSS to a simpler format.
☑	XMLtoHTML	iWay	none

The table that is provided lists all the transforms that are currently available in the registry.

6. Select a defined transform and click *Finish*.

You are returned to the Construct Transformer pane.



Transformers / SampleTransformer

Transformers are sequences of exits that manipulate a message as it passes thru a channel. Transformers can be applied before and after a process.

Construct Transformer

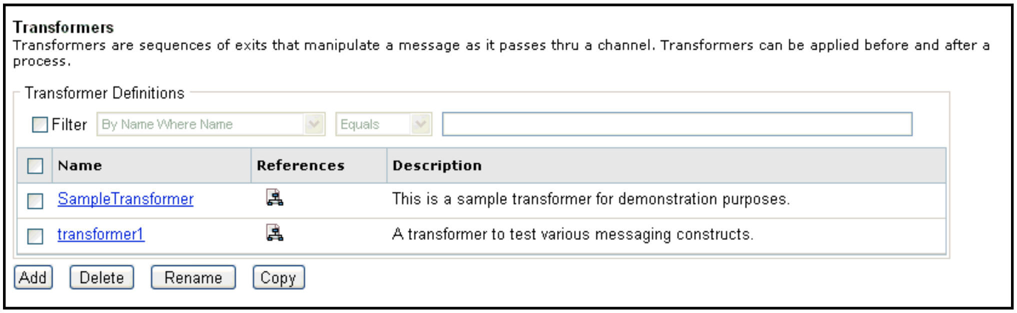
Below are the components currently registered in the transformer. The order of components may be changed within each component type by checking a component and using the 'Move Up' and 'Move Down' buttons.

Display as In Transformer

□	Name	Type	Move	Description
□	XMLtoHTML	Transform		none

7. Return to the main Transformers pane.

   The new transformer you defined is added.



Transformers

Transformers are sequences of exits that manipulate a message as it passes thru a channel. Transformers can be applied before and after a process.

Transformer Definitions

□ Filter By Name Where Name Equals

□	Name	References	Description
□	SampleTransformer		This is a sample transformer for demonstration purposes.
□	transformer1		A transformer to test various messaging constructs.

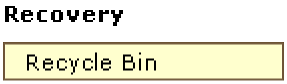
After a transformer is defined to iWay Service Manager, you can add it to a route. For more information, see *Configuring Channels* on page 305.

Using the Recycle Bin

The iWay Service Manager Administration Console provides a recycle bin feature, which allows you to restore or permanently destroy any iWay component that has been previously deleted from the registry. For example, if you inadvertently delete a listener, you can recover it using the recycle bin feature.

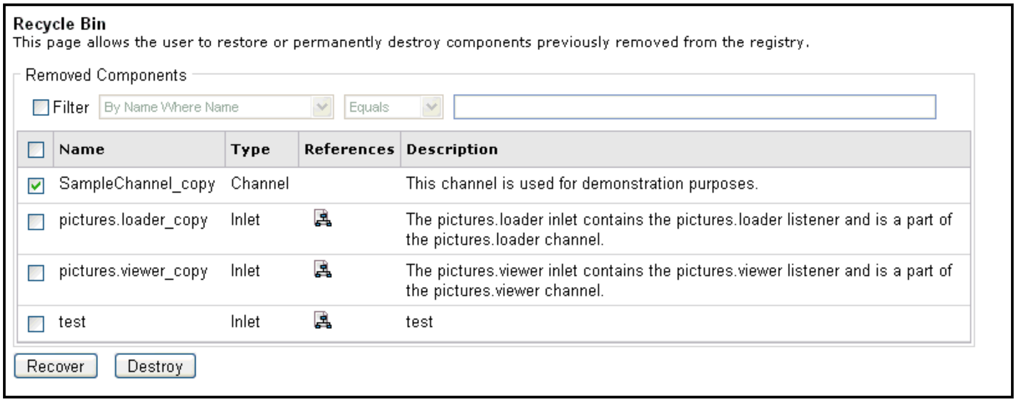
Procedure: How to Permanently Destroy or Restore iWay Components

To permanently destroy or restore iWay components:



1. In the left console pane of the Registry menu, select *Recycle Bin*.

The Recycle Bin pane opens.



Recycle Bin

This page allows the user to restore or permanently destroy components previously removed from the registry.

Removed Components

□ Filter By Name Where Name | Equals

□	Name	Type	References	Description
☑	SampleChannel_copy	Channel		This channel is used for demonstration purposes.
□	pictures.loader_copy	Inlet	品	The pictures.loader inlet contains the pictures.loader listener and is a part of the pictures.loader channel.
□	pictures.viewer_copy	Inlet	品	The pictures.viewer inlet contains the pictures.viewer listener and is a part of the pictures.viewer channel.
□	test	Inlet	品	test

The table that is provided lists each type of iWay component that has been deleted from the registry. It is a good practice to check for any applicable references a component may have before continuing. Click the schematic icon in the References column to view the references for an iWay component.

2. Click the check box next to the component you wish to permanently destroy or restore, for example, SampleChannel_copy.

   Perform one of the following steps:

   a. To permanently destroy an iWay component, click *Destroy*.

      A confirmation dialog box opens, prompting you to confirm the deletion of the iWay component.

      Click *OK*.

      The iWay component is permanently removed from the system.

   b. To restore an iWay component, click *Recover*.

      The iWay component is returned to the registry and can be used with iWay Service Manager.

      To permanently destroy or restore more than one component at once, you can select multiple check boxes in the Recycle Bin pane and click *Destroy* or *Recover*.

      Note: If you delete an item (for example, a channel) and then create a new one using the same name, the Recover action will result in two channels in the iWay Registry.

Chapter 8

Configuring Channels

The concept of a channel has been introduced to assist in the construction of message flows in iWay Service Manager. A channel serves as a container for all your components that simplifies the integration design process and improves organization, versioning, and troubleshooting. A channel contains the following conduits that must be configured and associated with the channel.

❑ **Inlet** - Defines how a message enters a channel.

❑ **Route** - Defines the path a message takes through a channel.

❑ **Outlet** - Defines how a message leaves a channel.

In this chapter:

Defining an Inlet

Each inlet contains a sequence of listeners, optional decryptors, and optional preparsers.

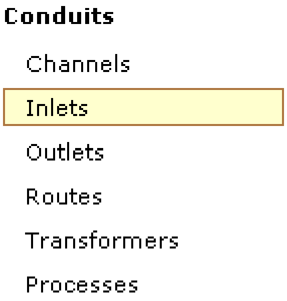
Listeners are defined as protocol handlers and are responsible for startup, shutdown, and obtaining the incoming messages. Listeners receive the messages from the transport protocol, set special registers such as header values and input source, and then pass the message to a decryptor.

Decryptors can apply a decryption algorithm to the incoming message and verify the security of the message. A decryptor can be used to verify that the sender is authorized, to check that the message has not been changed, and to decrypt any part of the message that has been encrypted. Finally, the decryptor passes the message to a preparser.

Preparsers convert transported messages into processable documents. For example, some preparsers convert non-XML messages into XML documents. Preparsers can be chained, so that the output of one preparser becomes the input of the next preparser. Input to the first preparser is a byte stream, and output is a properly encoded string. Subsequent preparsers accept and emit strings.

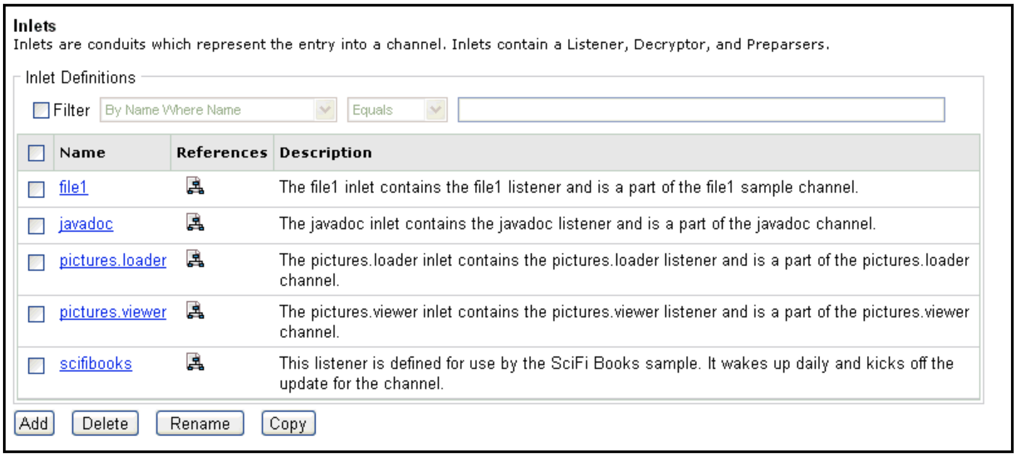
Procedure: How to Define an Inlet

To define an inlet using the iWay Service Manager Administration Console:



1. In the left console pane of the Registry menu, select *Inlets*.

   The Inlets pane opens.



Inlets

Inlets are conduits which represent the entry into a channel. Inlets contain a Listener, Decryptor, and Preparsers.

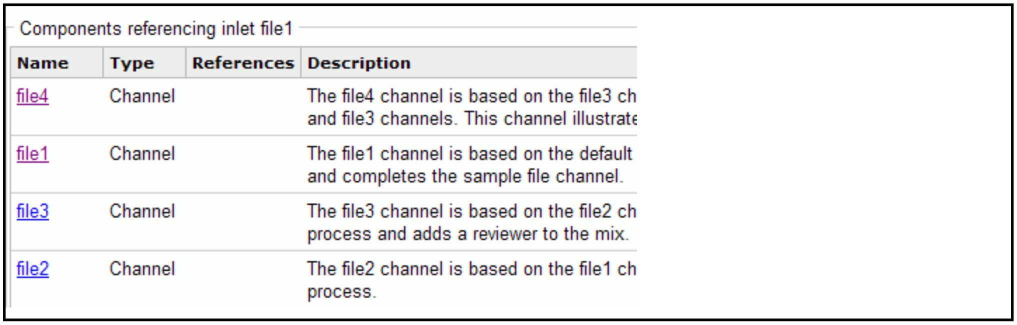
Inlet Definitions

□ Filter By Name Where Name | Equals |

□	Name	References	Description
□	file1	🖧	The file1 inlet contains the file1 listener and is a part of the file1 sample channel.
□	javadoc	🖧	The javadoc inlet contains the javadoc listener and is a part of the javadoc channel.
□	pictures.loader	🖧	The pictures.loader inlet contains the pictures.loader listener and is a part of the pictures.loader channel.
□	pictures.viewer	🖧	The pictures.viewer inlet contains the pictures.viewer listener and is a part of the pictures.viewer channel.
□	scifibooks	🖧	This listener is defined for use by the SciFi Books sample. It wakes up daily and kicks off the update for the channel.

The table that is provided lists each inlet that is defined with a brief description. If you click the document schematic icon in the References column for a specific inlet, you will see which components are referencing that inlet.

The following image shows the result of clicking the schematic icon for the file1 inlet:

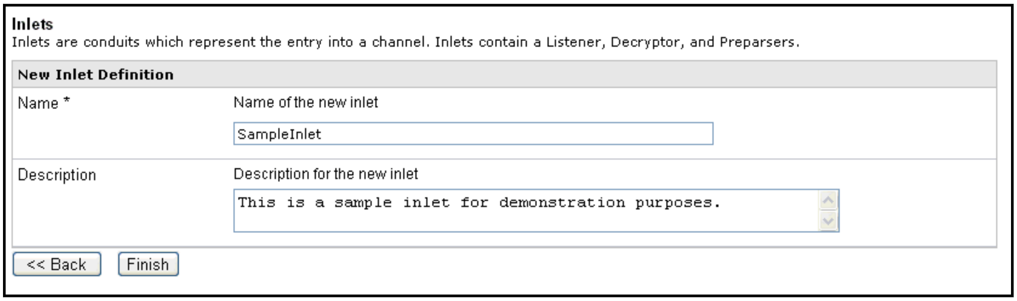


Components referencing inlet file1

Name	Type	References	Description
file4	Channel		The file4 channel is based on the file3 ch and file3 channels. This channel illustrate
file1	Channel		The file1 channel is based on the default and completes the sample file channel.
file3	Channel		The file3 channel is based on the file2 ch process and adds a reviewer to the mix.
file2	Channel		The file2 channel is based on the file1 ch process.

2. Click *Add*.

   The New Inlet Definition pane opens.



Inlets

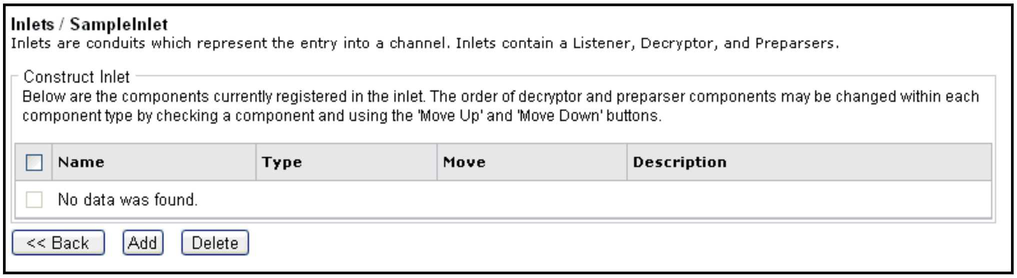
Inlets are conduits which represent the entry into a channel. Inlets contain a Listener, Decryptor, and Preparsers.

New Inlet Definition

Name *	Name of the new inlet	SampleInlet
Description	Description for the new inlet	This is a sample inlet for demonstration purposes.

3. Enter a name, for example, SampleInlet, and description for the inlet.
4. Click *Finish*.

   The Construct Inlet pane opens.



Inlets / SampleInlet

Inlets are conduits which represent the entry into a channel. Inlets contain a Listener, Decryptor, and Preparsers.

Construct Inlet

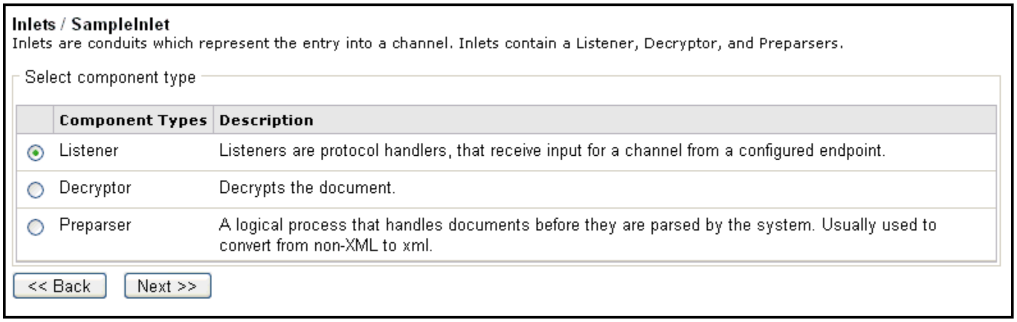
Below are the components currently registered in the inlet. The order of decryptor and preparser components may be changed within each component type by checking a component and using the 'Move Up' and 'Move Down' buttons.

□	Name	Type	Move	Description
□	No data was found.			

The table that is provided is used to list the components that are currently registered with the inlet.

5. Click *Add*.

   The Select component type pane opens.



Inlets / SampleInlet
Inlets are conduits which represent the entry into a channel. Inlets contain a Listener, Decryptor, and Preparsers.

Select component type

	Component Types	Description
●	Listener	Listeners are protocol handlers, that receive input for a channel from a configured endpoint.
○	Decryptor	Decrypts the document.
○	Preparser	A logical process that handles documents before they are parsed by the system. Usually used to convert from non-XML to xml.

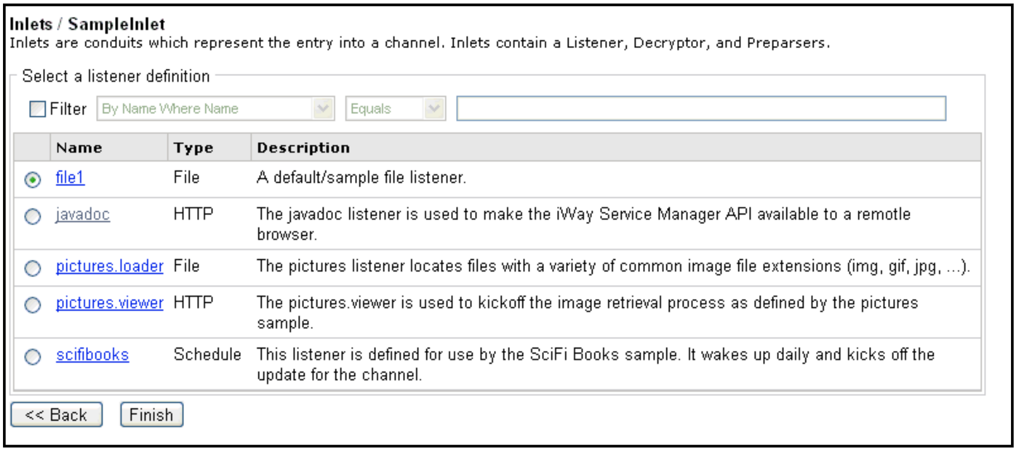
The table that is provided lists the component types you can select and register with the inlet you are defining.

- **Listener** - Protocol handlers that receive input for a channel from a configured endpoint.
- **Decryptor** - Used to decrypt a document.
- **Preparser** - A logical process that handles documents before they are parsed by the system, for example, converting a non-XML document to XML.

Note: Each inlet is required to have a registered listener. The remaining components are optional during inlet configuration. For more information on creating a listener, see *Configuring iWay Registry Components* on page 249.

6. Select *Listener* from the list of component types and click *Next*.

   The Select a listener definition pane opens, as shown in the following image.



Inlets / SampleInlet
Inlets are conduits which represent the entry into a channel. Inlets contain a Listener, Decryptor, and Preparsers.

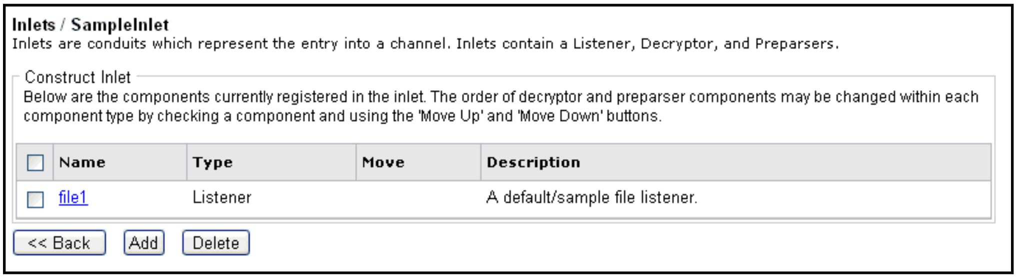
Select a listener definition

□ Filter By Name Where Name Equals

	Name	Type	Description
●	file1	File	A default/sample file listener.
○	javadoc	HTTP	The javadoc listener is used to make the iWay Service Manager API available to a remotle browser.
○	pictures.loader	File	The pictures listener locates files with a variety of common image file extensions (img, gif, jpg, ...).
○	pictures.viewer	HTTP	The pictures.viewer is used to kickoff the image retrieval process as defined by the pictures sample.
○	scifibooks	Schedule	This listener is defined for use by the SciFi Books sample. It wakes up daily and kicks off the update for the channel.

7. Select a defined listener, for example, file1, from the list and click *Finish*.

   You are returned to the Construct Inlet pane, which now includes the listener (file1) you registered with your inlet (SampleInlet).



Inlets / SampleInlet

Inlets are conduits which represent the entry into a channel. Inlets contain a Listener, Decryptor, and Preparsers.

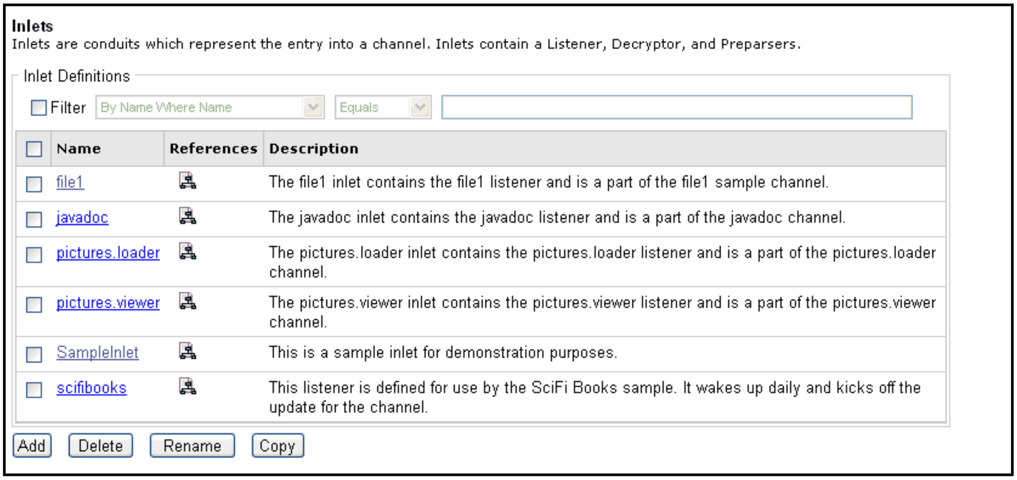
Construct Inlet

Below are the components currently registered in the inlet. The order of decryptor and preparser components may be changed within each component type by checking a component and using the 'Move Up' and 'Move Down' buttons.

□	Name	Type	Move	Description
□	file1	Listener		A default/sample file listener.

You can repeat this process to define additional components for the inlet, such as a decryptor or a preparser.

If you return to the main Inlets pane, you will notice that the inlet you just defined (SampleInlet) has been added to the list, as shown in the following image.



Inlets

Inlets are conduits which represent the entry into a channel. Inlets contain a Listener, Decryptor, and Preparsers.

Inlet Definitions

□ Filter By Name Where Name Equals

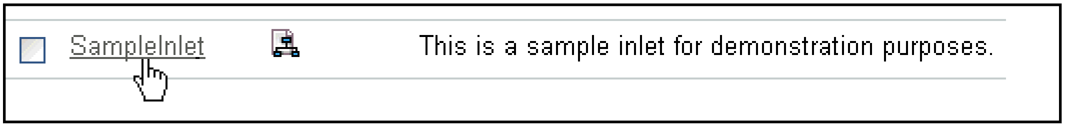
□	Name	References	Description
□	file1	🖧	The file1 inlet contains the file1 listener and is a part of the file1 sample channel.
□	javadoc	🖧	The javadoc inlet contains the javadoc listener and is a part of the javadoc channel.
□	pictures.loader	🖧	The pictures.loader inlet contains the pictures.loader listener and is a part of the pictures.loader channel.
□	pictures.viewer	🖧	The pictures.viewer inlet contains the pictures.viewer listener and is a part of the pictures.viewer channel.
□	SampleInlet	🖧	This is a sample inlet for demonstration purposes.
□	scifibooks	🖧	This listener is defined for use by the SciFi Books sample. It wakes up daily and kicks off the update for the channel.

Procedure: How to Modify an Inlet

You can modify an inlet by changing one of the components already defined for the inlet or by adding new components.

Note: If you change one of the components assigned to an inlet, such as a listener, the component will be changed in the registry, not only in the particular inlet you are modifying.

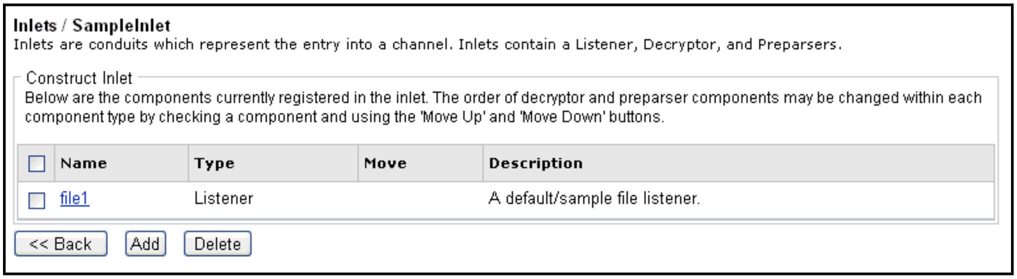
To modify an inlet:



□ SampleInlet This is a sample inlet for demonstration purposes.

1. Click the name of the inlet you wish to modify in the main Inlets pane, for example, SampleInlet.

   The Construct Inlet pane opens.



Inlets / SampleInlet
Inlets are conduits which represent the entry into a channel. Inlets contain a Listener, Decryptor, and Preparsers.

Construct Inlet
Below are the components currently registered in the inlet. The order of decryptor and preparser components may be changed within each component type by checking a component and using the 'Move Up' and 'Move Down' buttons.

□	Name	Type	Move	Description
□	file1	Listener		A default/sample file listener.

   The table that is provided shows the component that has been registered for the inlet you are modifying. In this example, the file1 listener is used.

2. Modify the inlet as required:

   ❑ To replace one of the components, such as a listener, remove the name of the listener.

   ❑ To add a new component, select the component type, and click *Next*.

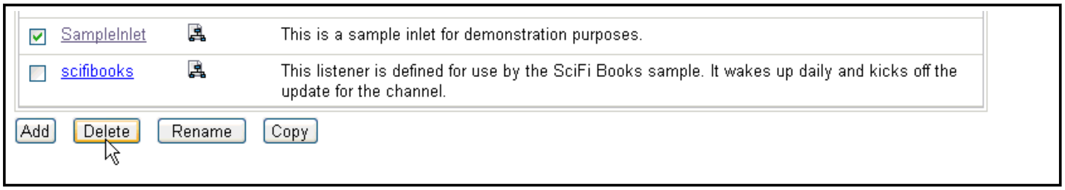
   For more information on listeners, decryptors, and preparsers, see *Configuring iWay Registry Components* on page 249.

3. Make your changes to the listener properties as required and click *Update* when you are finished, or select the component you are adding and click *Finish*.

   The inlet is now modified.

Procedure: How to Delete an Inlet

To delete an inlet:



☑	SampleInlet	This is a sample inlet for demonstration purposes.
□	scifibooks	This listener is defined for use by the SciFi Books sample. It wakes up daily and kicks off the update for the channel.

1. Click the check box next to the inlet you wish to delete, for example, SampleInlet.
2. Click *Delete*.

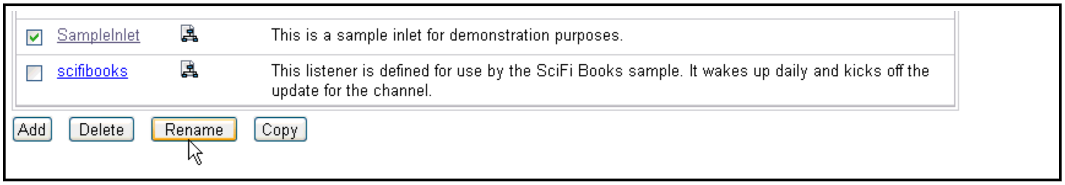
   A confirmation dialog box opens, prompting you to confirm the deletion of the inlet.
3. Click *OK*.

   The inlet is deleted from the system.

   Tip: To delete more than one inlet at once, you can select multiple check boxes in the Inlets pane and click *Delete*.

Procedure: How to Rename an Inlet

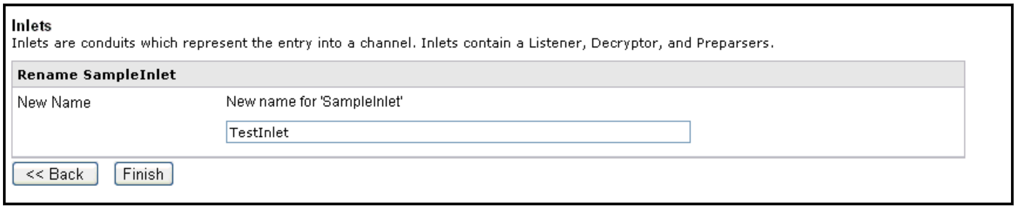
To rename an inlet:



☑	SampleInlet	This is a sample inlet for demonstration purposes.
□	scifibooks	This listener is defined for use by the SciFi Books sample. It wakes up daily and kicks off the update for the channel.

1. Click the check box next to the inlet you wish to rename, for example, SampleInlet.
2. Click *Rename*.

   The Rename pane opens.



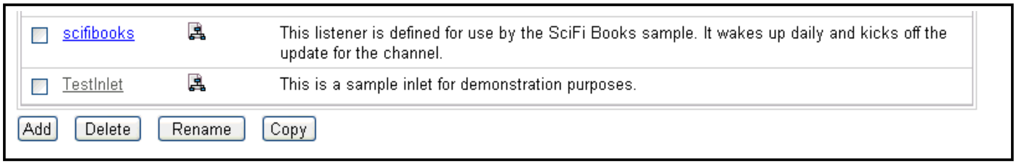
Inlets

Inlets are conduits which represent the entry into a channel. Inlets contain a Listener, Decryptor, and Preparsers.

Rename SampleInlet	
New Name	New name for 'SampleInlet' TestInlet

3. Type a name, for example, TestInlet, in the New Name field and click *Finish*.

You are returned to the main Inlets pane.

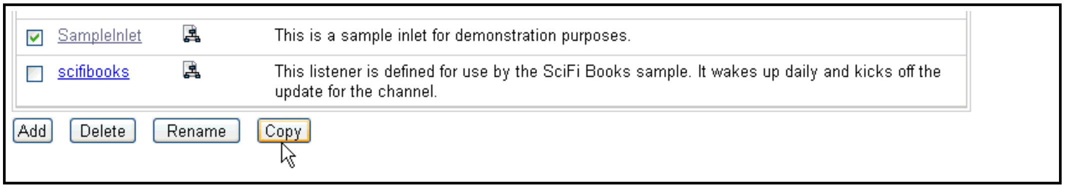


□	scifibooks	This listener is defined for use by the SciFi Books sample. It wakes up daily and kicks off the update for the channel.
□	TestInlet	This is a sample inlet for demonstration purposes.

The new name for the inlet you provided is shown.

Procedure: How to Copy an Inlet

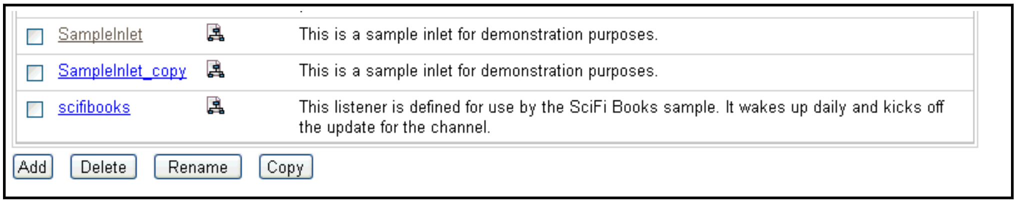
To copy an inlet:



☑	SampleInlet	This is a sample inlet for demonstration purposes.
□	scifibooks	This listener is defined for use by the SciFi Books sample. It wakes up daily and kicks off the update for the channel.

1. Click the check box next to the inlet you wish to copy, for example, SampleInlet.
2. Click *Copy*.

   A copy of the inlet is made, as shown in the following image.



□	SampleInlet	This is a sample inlet for demonstration purposes.
□	SampleInlet_copy	This is a sample inlet for demonstration purposes.
□	scifibooks	This listener is defined for use by the SciFi Books sample. It wakes up daily and kicks off the update for the channel.

Making copies of an inlet is useful for versioning and testing purposes.

Tip: To copy more than one inlet at once, you can select multiple check boxes in the Inlets pane and click *Copy*.

Defining a Route

Routes contain references to transformers, processes, and outlets. A route describes the path that a document takes during its passage through the system, after the inlet converts the input message to a document that can be processed. Multiple routes can be defined for the same channel if required, but only one route is executed. The executed route is the first route in the processing order that meets the condition criteria for the document or is a default route if no condition is met.

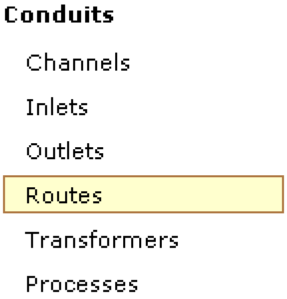
Transformers contain a transform component such as information to a common format suitable for general business processing. For example, similar messages from two trading partners might differ slightly in format; a common format is often desirable for business processing. Transforms are constructed using iWay Transformer. For more information, see the *iWay Transformer User's Guide*.

Processes perform the actual business operations on the document. A business process is composed of one or more services, with appropriate switching, testing, iteration, and error handling. Processes can call on other processes and web services, and in turn can be packaged as web services for external consumption. Business processes are constructed using iWay Designer. For more information, see the *iWay Designer User's Guide*.

Outlets pass the processed document to one or more designated recipients. They convert the document to a transport format and then emit the message.

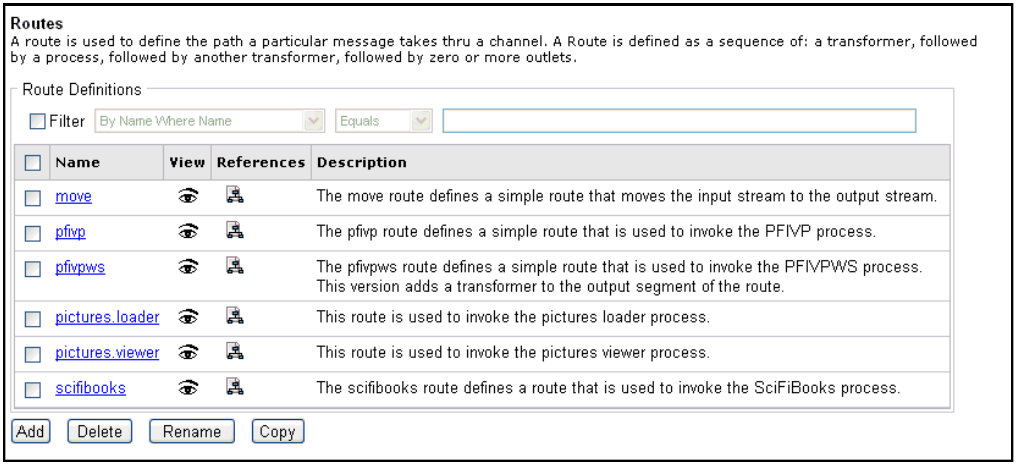
Procedure: How to Define a Route

To define a route using the iWay Service Manager Administration Console:



1. In the left console pane of the Registry menu, select *Routes*.

The Routes pane opens.



Routes

A route is used to define the path a particular message takes thru a channel. A Route is defined as a sequence of: a transformer, followed by a process, followed by another transformer, followed by zero or more outlets.

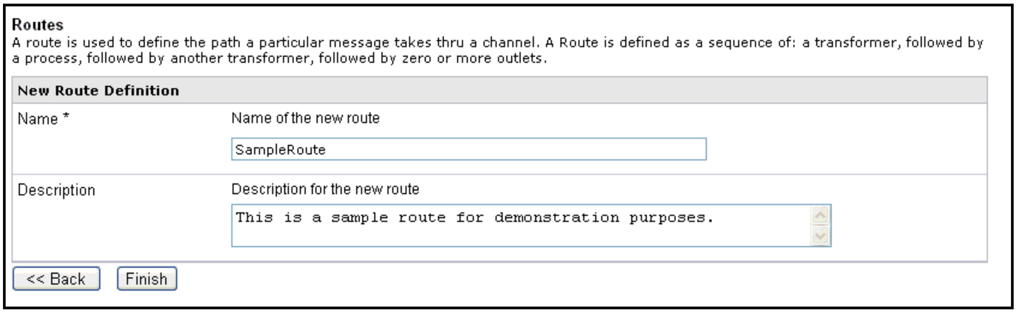
Route Definitions

□	Name	View	References	Description
□	move	👁	🖧	The move route defines a simple route that moves the input stream to the output stream.
□	pfivp	👁	🖧	The pfivp route defines a simple route that is used to invoke the PFIVP process.
□	pfivpws	👁	🖧	The pfivpws route defines a simple route that is used to invoke the PFIVPWS process. This version adds a transformer to the output segment of the route.
□	pictures.loader	👁	🖧	This route is used to invoke the pictures loader process.
□	pictures.viewer	👁	🖧	This route is used to invoke the pictures viewer process.
□	scifibooks	👁	🖧	The scifibooks route defines a route that is used to invoke the SciFiBooks process.

The table that is provided lists each route that is defined with a brief description. If you click the document schematic icon in the References column for a specific route, you will see which components are referencing that route. (If you click the eye icon in the View column for a specific route, and then the process icon between the two arrows, you will see a visual depiction of that route.)

2. Click *Add*.

   The New Route Definition pane opens.



Routes

A route is used to define the path a particular message takes thru a channel. A Route is defined as a sequence of: a transformer, followed by a process, followed by another transformer, followed by zero or more outlets.

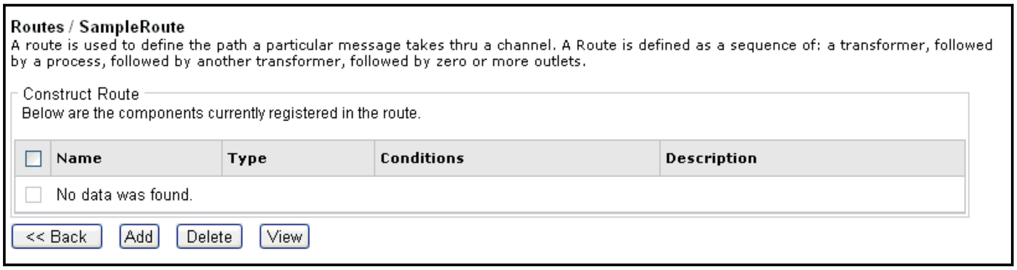
New Route Definition

Name *	Name of the new route	SampleRoute
Description	Description for the new route	This is a sample route for demonstration purposes.

3. Enter a name, for example, SampleRoute, and description for the route.

4. Click *Finish*.

The Construct Route pane opens.



Routes / SampleRoute

A route is used to define the path a particular message takes thru a channel. A Route is defined as a sequence of: a transformer, followed by a process, followed by another transformer, followed by zero or more outlets.

Construct Route

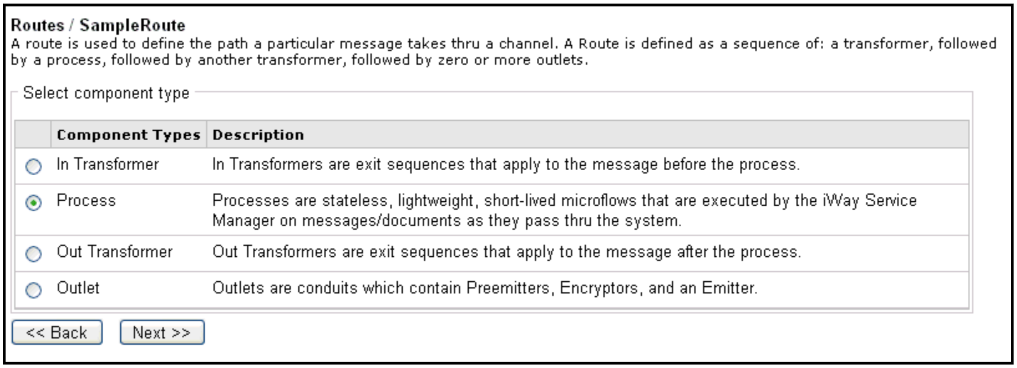
Below are the components currently registered in the route.

□	Name	Type	Conditions	Description
□	No data was found.			

The table that is provided is used to list the components that are currently registered with the route.

5. Click *Add*.

   The Select component type pane opens.



Routes / SampleRoute

A route is used to define the path a particular message takes thru a channel. A Route is defined as a sequence of: a transformer, followed by a process, followed by another transformer, followed by zero or more outlets.

Select component type

	Component Types	Description
○	In Transformer	In Transformers are exit sequences that apply to the message before the process.
●	Process	Processes are stateless, lightweight, short-lived microflows that are executed by the iWay Service Manager on messages/documents as they pass thru the system.
○	Out Transformer	Out Transformers are exit sequences that apply to the message after the process.
○	Outlet	Outlets are conduits which contain Preemitters, Encryptors, and an Emitter.

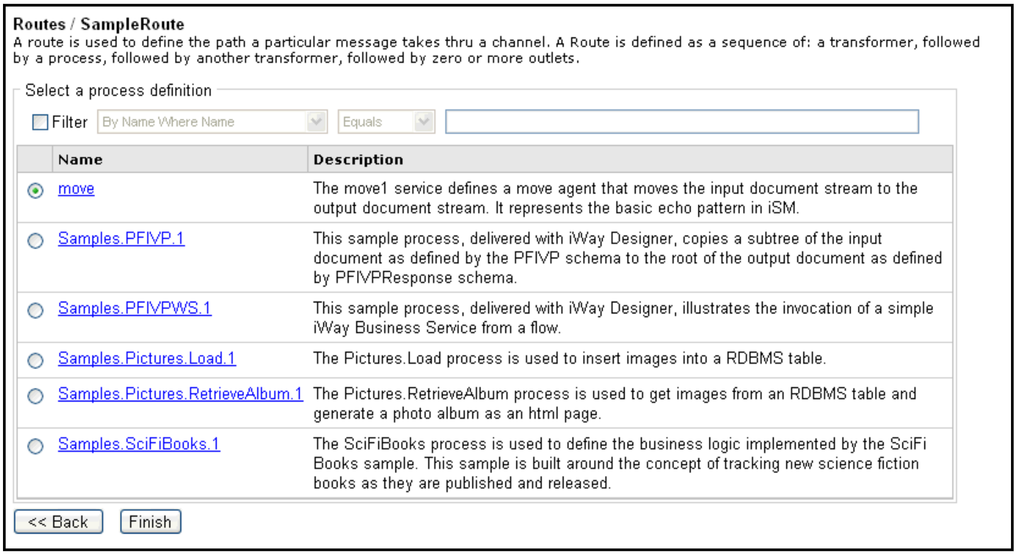
The table that is provided lists the component types you can select and register with the route you are defining.

- **In Transformer** - Exit sequences that apply to a message before processing occurs.
- **Process** - Stateless, lightweight, and short-lived microflows that are executed by iWay Service Manager on messages and documents as they pass through the system. The simplest process contains a move service, which is first placed into a flow then added to a process.
- **Out Transformer** - Exit sequences that apply to the message after processing occurs.
- **Outlet** - Conduits that consist of Preemitters, Encryptors, and Emitters.

Note: Each route that is being defined is required to have a registered process. The remaining components are optional during route configuration. For more information on creating a component, see *Configuring iWay Registry Components* on page 249.

6. Select *Process* from the list of component types and click *Next*.

   The Select a process definition pane opens.



Routes / SampleRoute
A route is used to define the path a particular message takes thru a channel. A Route is defined as a sequence of: a transformer, followed by a process, followed by another transformer, followed by zero or more outlets.

Select a process definition

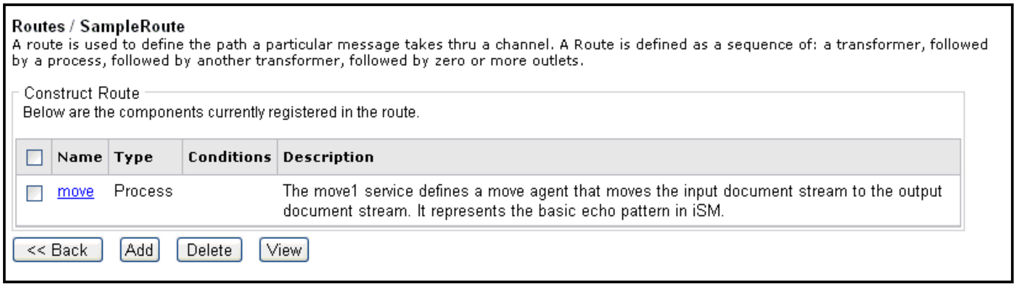
□ Filter | By Name Where Name | Equals |

	Name	Description
●	move	The move1 service defines a move agent that moves the input document stream to the output document stream. It represents the basic echo pattern in iSM.
○	Samples.PFIVP.1	This sample process, delivered with iWay Designer, copies a subtree of the input document as defined by the PFIVP schema to the root of the output document as defined by PFIVPResponse schema.
○	Samples.PFIVPWS.1	This sample process, delivered with iWay Designer, illustrates the invocation of a simple iWay Business Service from a flow.
○	Samples.Pictures.Load.1	The Pictures.Load process is used to insert images into a RDBMS table.
○	Samples.Pictures.RetrieveAlbum.1	The Pictures.RetrieveAlbum process is used to get images from an RDBMS table and generate a photo album as an html page.
○	Samples.SciFiBooks.1	The SciFiBooks process is used to define the business logic implemented by the SciFi Books sample. This sample is built around the concept of tracking new science fiction books as they are published and released.

   The table that is provided lists existing process flows you can select for the route you are defining.

7. Select *move* and click *Finish*.

   You are returned to the Construct Route pane, which now includes the process (move) you registered with your route (SampleRoute).



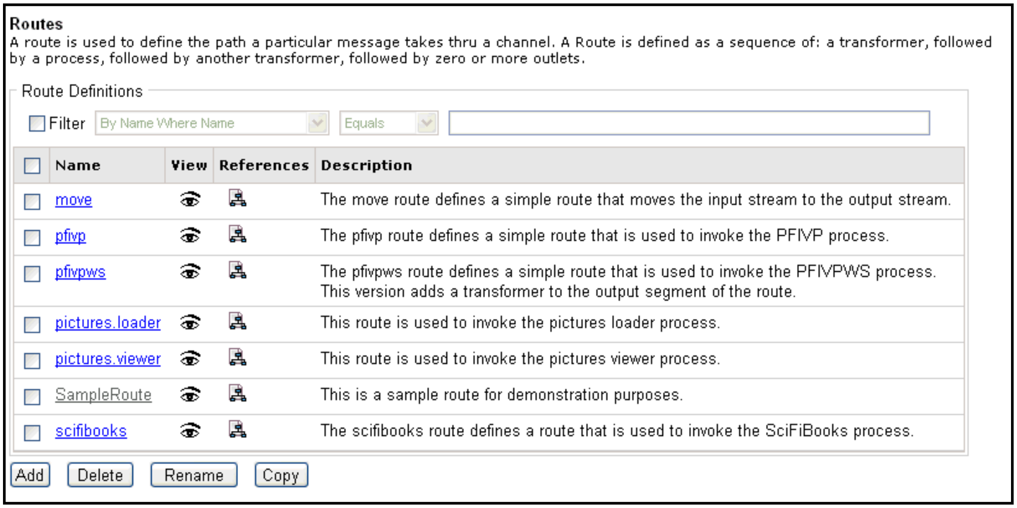
Routes / SampleRoute
A route is used to define the path a particular message takes thru a channel. A Route is defined as a sequence of: a transformer, followed by a process, followed by another transformer, followed by zero or more outlets.

Construct Route
Below are the components currently registered in the route.

□	Name	Type	Conditions	Description
□	move	Process		The move1 service defines a move agent that moves the input document stream to the output document stream. It represents the basic echo pattern in iSM.

   You can now add additional components, such as a transformer or an outlet.

If you return to the main Routes pane, you will notice that the route you just defined (SampleRoute) has been added to the list, as shown in the following image.



Routes

A route is used to define the path a particular message takes thru a channel. A Route is defined as a sequence of: a transformer, followed by a process, followed by another transformer, followed by zero or more outlets.

Route Definitions

☐ Filter By Name Where Name Equals

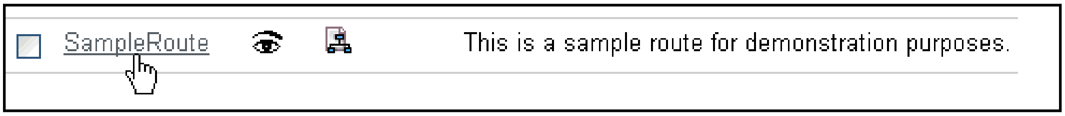
☐	Name	View	References	Description
☐	move	👁	🖧	The move route defines a simple route that moves the input stream to the output stream.
☐	pfivp	👁	🖧	The pfivp route defines a simple route that is used to invoke the PFIVP process.
☐	pfivpws	👁	🖧	The pfivpws route defines a simple route that is used to invoke the PFIVPWS process. This version adds a transformer to the output segment of the route.
☐	pictures.loader	👁	🖧	This route is used to invoke the pictures loader process.
☐	pictures.viewer	👁	🖧	This route is used to invoke the pictures viewer process.
☐	SampleRoute	👁	🖧	This is a sample route for demonstration purposes.
☐	scifibooks	👁	🖧	The scifibooks route defines a route that is used to invoke the SciFiBooks process.

Procedure: How to Modify a Route

You can modify a route by changing one of the components already defined for the route or by adding new components.

Note: If you change one of the components assigned to a route, such as a process, the component will be changed in the registry, not only in the particular route you are modifying.

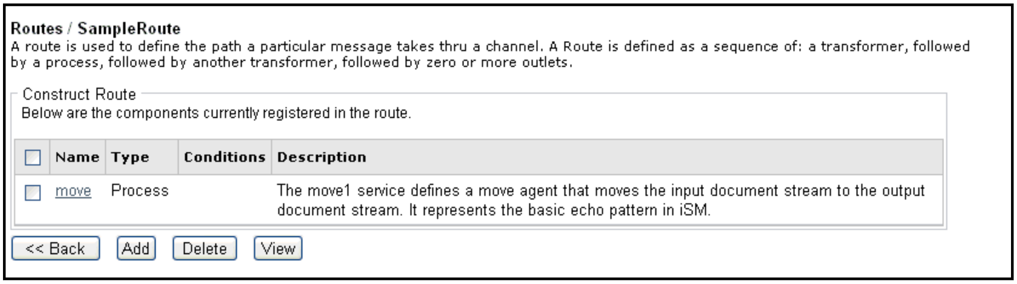
To modify a route:



☐ SampleRoute 👁 🖧 This is a sample route for demonstration purposes.

1. Click the name of the route you wish to modify in the main Routes pane, for example, SampleRoute.

The Construct Route pane opens.



Routes / SampleRoute
A route is used to define the path a particular message takes thru a channel. A Route is defined as a sequence of: a transformer, followed by a process, followed by another transformer, followed by zero or more outlets.

Construct Route
Below are the components currently registered in the route.

□	Name	Type	Conditions	Description
□	move	Process		The move1 service defines a move agent that moves the input document stream to the output document stream. It represents the basic echo pattern in iSM.

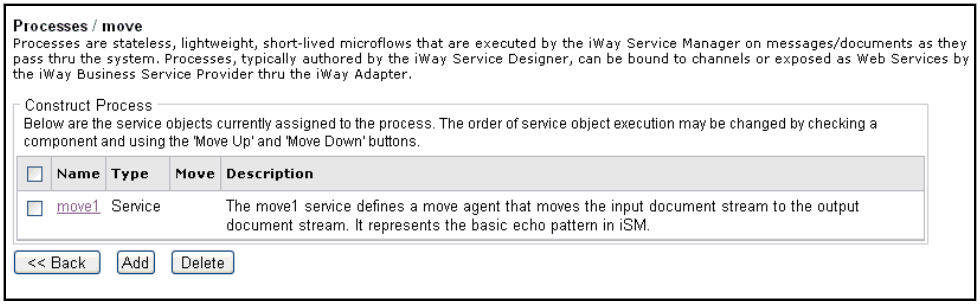
The table that is provided shows the component that has been registered for the route you are modifying. In this example, the move process is used.

2. Modify the route as required:

   The table that is provided lists components that are currently assigned to the selected process.

   ❑ To change one of the components already defined, such as a process, in this case the move process, click the name of the component.

   The Construct Process pane for the move process opens.



Processes / move
Processes are stateless, lightweight, short-lived microflows that are executed by the iWay Service Manager on messages/documents as they pass thru the system. Processes, typically authored by the iWay Service Designer, can be bound to channels or exposed as Web Services by the iWay Business Service Provider thru the iWay Adapter.

Construct Process
Below are the service objects currently assigned to the process. The order of service object execution may be changed by checking a component and using the 'Move Up' and 'Move Down' buttons.

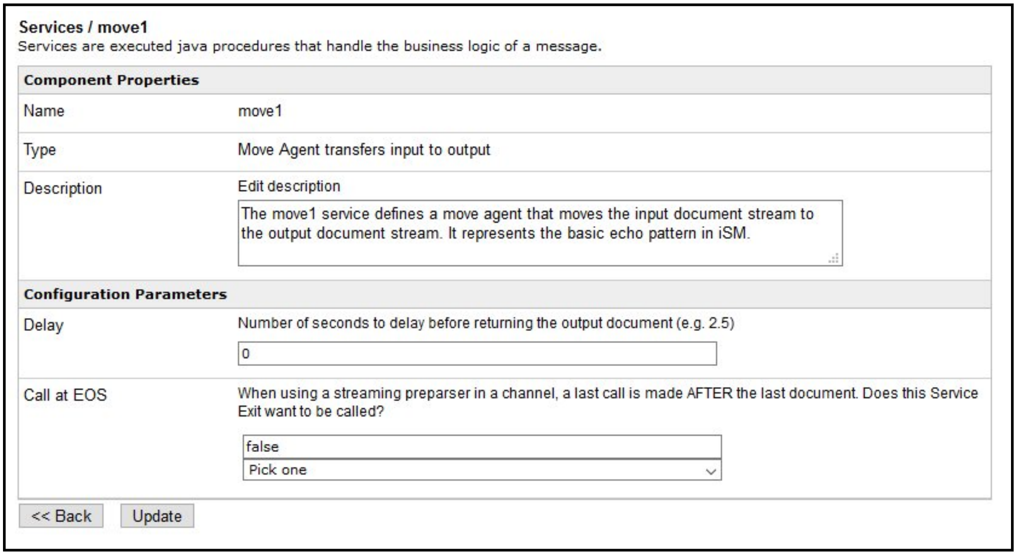
□	Name	Type	Move	Description
□	move1	Service		The move1 service defines a move agent that moves the input document stream to the output document stream. It represents the basic echo pattern in iSM.

   ❑ To add a new component, such as an In Transformer, an Out Transformer, or an Outlet, select the component type and click *Next*.

   For more information on transforms, see *Configuring iWay Registry Components* on page 249.

3. Click the *move1* service.

The Component Properties pane opens.



Services / move1

Services are executed java procedures that handle the business logic of a message.

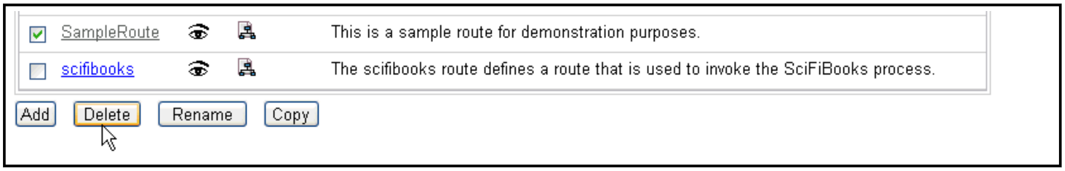
Component Properties	
Name	move1
Type	Move Agent transfers input to output
Description	Edit description The move1 service defines a move agent that moves the input document stream to the output document stream. It represents the basic echo pattern in iSM.
Configuration Parameters	
Delay	Number of seconds to delay before returning the output document (e.g. 2.5) 0
Call at EOS	When using a streaming preparser in a channel, a last call is made AFTER the last document. Does this Service Exit want to be called? false Pick one

4. Make your changes to the service properties as required and click *Update* or select the component you wish to add and click *Finish*.

   The route is now modified.

Procedure: How to Delete a Route

To delete a route:



☑	SampleRoute		This is a sample route for demonstration purposes.
□	scifibooks		The scifibooks route defines a route that is used to invoke the SciFiBooks process.

1. Click the check box next to the route you wish to delete, for example, SampleRoute.
2. Click *Delete*.

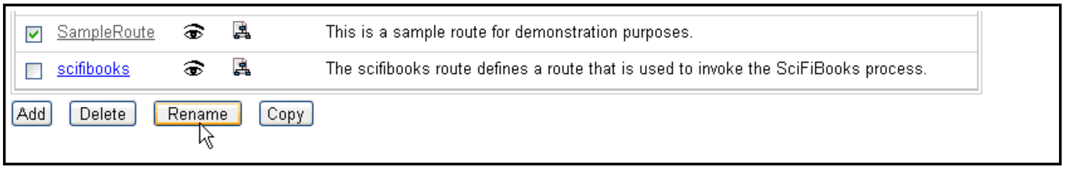
   A confirmation dialog box opens, prompting you to confirm the deletion of the route.
3. Click *OK*.

   The route is deleted from the system.

   Tip: To delete more than one route at once, you can select multiple check boxes in the Routes pane and click *Delete*.

Procedure: How to Rename a Route

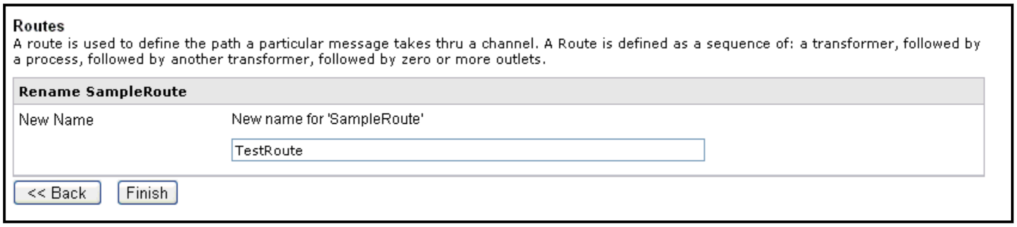
To rename a route:



☑	SampleRoute	This is a sample route for demonstration purposes.
□	scifibooks	The scifibooks route defines a route that is used to invoke the SciFiBooks process.

1. Click the check box next to the route you wish to rename, for example, SampleRoute.
2. Click *Rename*.

   The Rename pane opens.



Routes

A route is used to define the path a particular message takes thru a channel. A Route is defined as a sequence of: a transformer, followed by a process, followed by another transformer, followed by zero or more outlets.

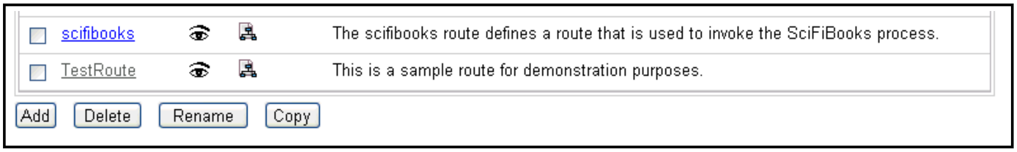
Rename SampleRoute

New Name: New name for 'SampleRoute'

TestRoute

3. Type a name, for example, TestRoute, in the New Name field and click *Finish*.

   You are returned to the main Routes pane.

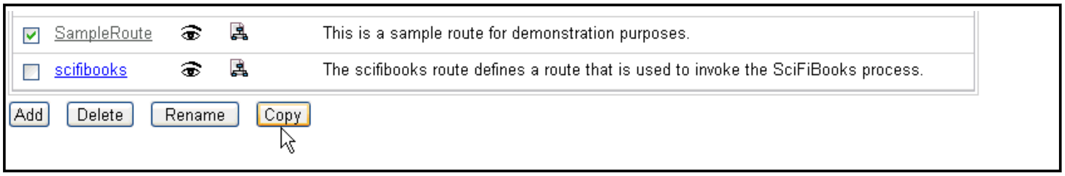


□	scifibooks	The scifibooks route defines a route that is used to invoke the SciFiBooks process.
□	TestRoute	This is a sample route for demonstration purposes.

The new name for the route you provided is shown.

Procedure: How to Copy a Route

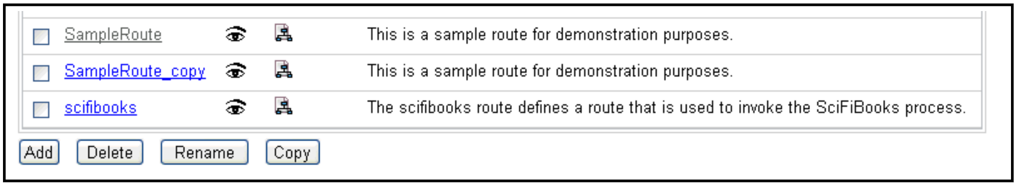
To copy a route:



☑	SampleRoute	This is a sample route for demonstration purposes.
□	scifibooks	The scifibooks route defines a route that is used to invoke the SciFiBooks process.

1. Click the check box next to the route you wish to copy, for example, SampleRoute.
2. Click *Copy*.

A copy of the route is made, as shown in the following image.



□	SampleRoute			This is a sample route for demonstration purposes.
□	SampleRoute_copy			This is a sample route for demonstration purposes.
□	scifibooks			The scifibooks route defines a route that is used to invoke the SciFiBooks process.

Making copies of a route is useful for versioning and testing purposes.

Tip: To copy more than one route at once, you can select multiple check boxes in the Routes pane and click *Copy*.

Routing Strategies

iWay Service Manager provides mechanisms to support routing strategies. This section describes how you can use iSM to execute business logic and route documents to a particular location. The following topics are provided:

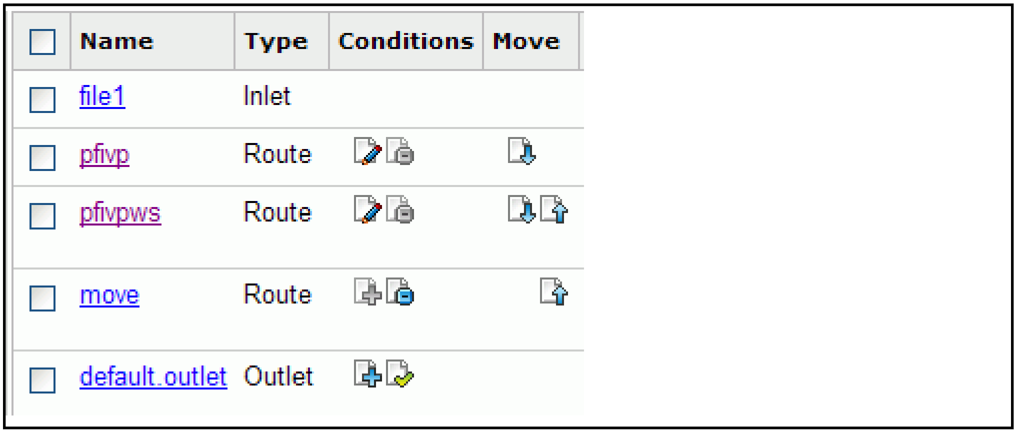
- ❑ Modifying the processing order when multiple routes are used in a channel.
- ❑ Setting default routes.
- ❑ Adding conditions to routes.

Procedure: How to Modify the Route Processing Order

The route processing order depends on how the routes are configured after they are added to a channel. The routing order can be modified by reordering the position of a route on the list. During runtime, only the first route that meets the condition or is a default route is executed. The ordering of the routes implies that the conditions of the routes will be checked based on the order. In this case, the first instance of the route to meet the condition will be executed or the default route will be executed.

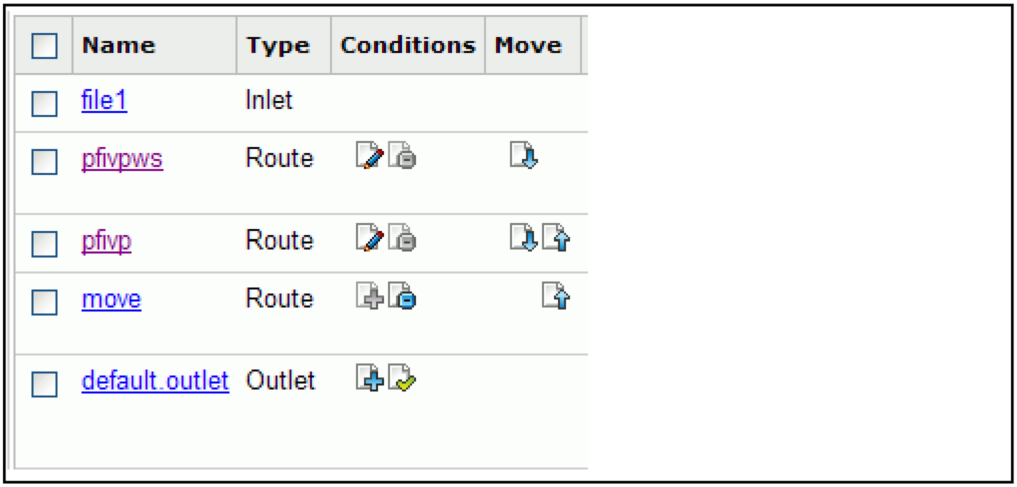
1. Open the channel you want to edit.
2. In the move channel, click either the up or down icon to modify the order of routes.

The following image shows the order of routes before the modification.



□	Name	Type	Conditions	Move
□	file1	Inlet		
□	pfivp	Route	✎	↓
□	pfivpws	Route	✎	↓ ↑
□	move	Route	+	↑
□	default.outlet	Outlet	+ ✓	

The following image shows the order of routes after the pfivpws route was moved up.



□	Name	Type	Conditions	Move
□	file1	Inlet		
□	pfivpws	Route	✎	↓
□	pfivp	Route	✎	↓ ↑
□	move	Route	+	↑
□	default.outlet	Outlet	+ ✓	

3. You can continue to edit the channel if necessary, or click *Build* to build the modified channel.

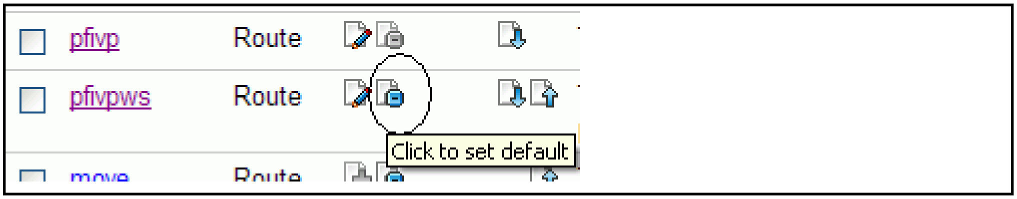
Procedure: How to Set Default Routes

As a best practice, set one of the routes as a default to ensure the document is processed if none of the specified conditions are met. If no default route is set, then a warning message is displayed when you build the channel and the document is not processed at run time. If conditions are assigned to the route(s) on a channel and if a document does not meet the assigned conditions, then the document is processed by the default route.

The placement of the default route matters if there are no conditions on the routes assigned to a channel.

1. Open the channel you want to edit.
2. Click the *set default* icon in the Conditions column in the route that you want to designate as the default.

   The following image shows the set default icon.



□	pfivp	Route
□	pfivpws	Route
□	move	Route

The set default icon on the route you modified now appears as active.

Conditional Routing

As a document moves through a channel, you can apply conditions to the routes assigned to the channel to direct the flow of the document dynamically based on the document's contents. The particular route that the document takes is therefore selected based on whether the document matches the conditions. If none of the conditions on the route(s) is met, the document will be handled by the default route. If no default route is set, the message will not be handled.

You can also control the routing of a document by specifying whether a channel allows fixed or dynamic routing. If the routing on a channel is set to fixed, a selected route in the channel remains in force for the duration of the messaging process. Dynamic routing, which is the default when a channel is created, allows the channel to evaluate the message at each stage in the document processing to determine which route to use. For more information on fixed and dynamic routing, see *Specifying Fixed or Dynamic Routing* on page 343.

The conditional routing and test feature supports complex expressions and a wide range of tests that can be performed. It compiles a complex expression the first time that it is encountered and uses the compilation in subsequent testing. This feature results in faster processing.

A requirement of conditional routing is that the function return a value of true or false, as do the COND(), ALL(), and ANY() functions as well as the _IS and _HAS functions.

You can use other functions that return a value and test their result in the COND(), ALL(), and ANY() function. Examples include the _LENGTH() and _SUBSTR() functions.

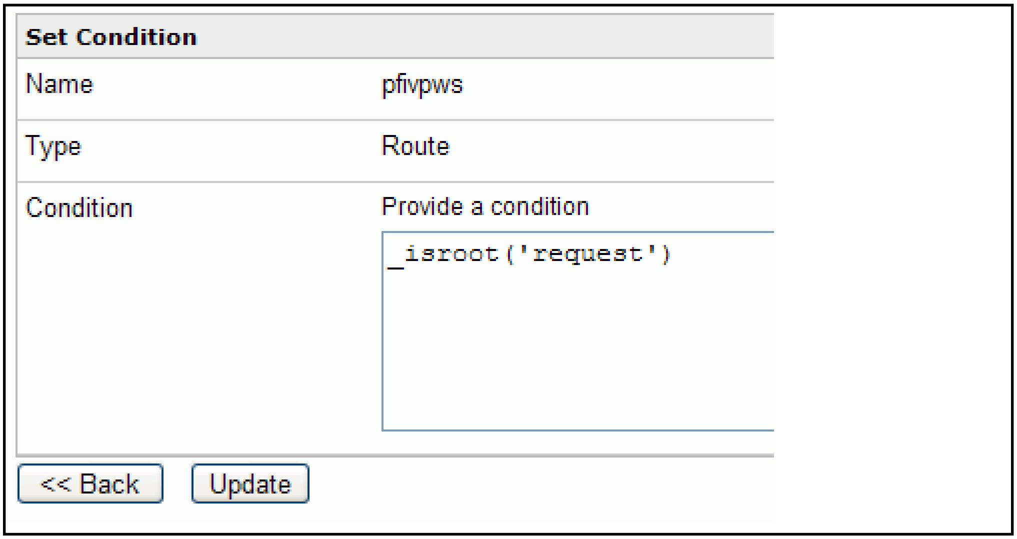
For more information on using functions, see the *iWay Service Manager Component and Functional Language Reference Guide*.

Procedure: How to Add or Edit Conditions on Routes

You add conditions to routes after the route has been added to a channel. In this way, you can reuse routes that have already been defined and customize them to perform the document processing required by each channel.

1. Open the channel you want to modify.
2. In the Conditions column for the route you want to modify, click the icon in the left of the column, either the icon with the plus sign or with the pencil, depending on whether conditions already exist for the route.

   The Set Condition pane appears, as shown in the following image.



Set Condition

Name	pfivpws
Type	Route
Condition	Provide a condition _isroot('request')

   If no condition is set, iWay Service Manager first checks the route tag in the document to see if it matches the route name.

3. Specify the condition and click *Update*.

   The route is updated with the condition.

4. Click *Back* to return to the channel pane.

Defining an Outlet

Outlets contain references to preemitters, encryptors, and emitters. Once a document has been processed, it must be sent to one or more designated recipients. This is the job of the outlet. The outlet is responsible for all aspects of preparing the document for emission and then emitting it.

Outlets contain a sequence of components tailored for this task. Multiple outlets can be configured for a single message flow. Each outlet incorporates all of the components needed to send the message to its destination.

A channel must contain an outlet; however, a default outlet, which contains no emitter, can be used. When you assign an empty outlet to a channel, the document output goes back to the listener assigned to the inlet and is emitted through whatever output is specified in the listener.

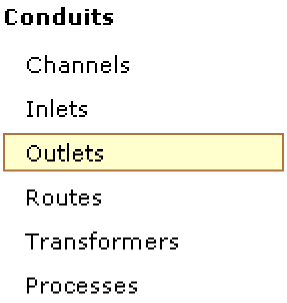
Preemitters convert the document from the internal format to an external format. This may include simply flattening XML or may be more complex, involving transformation logic. An example is converting the document to an EDI or HIPAA format. Preemitters can also be chained, so the output of one becomes the input to the next. The first preemitter receives the document in internal form, and transforms it to a message format. Subsequent preemitters can perform extra work on this message. Transformations are prepared using the iWay Transformer. For more information, see the *iWay Transformer User’s Guide*.

Encryptors operate on the message that is ready for emitting. Parameters such as the location of encryption keys or certificate aliases can be stored by destination address in the optional iWay Trading Manager component. For more information, see the *iWay Trading Manager User’s Guide*.

The emitter uses the appropriate transport protocol to send the document to its destination. Examples include JMS, HTTP, email, and AS2. Header information that has been prepared by processes and stored in special registers is applied to the message in a format-appropriate manner.

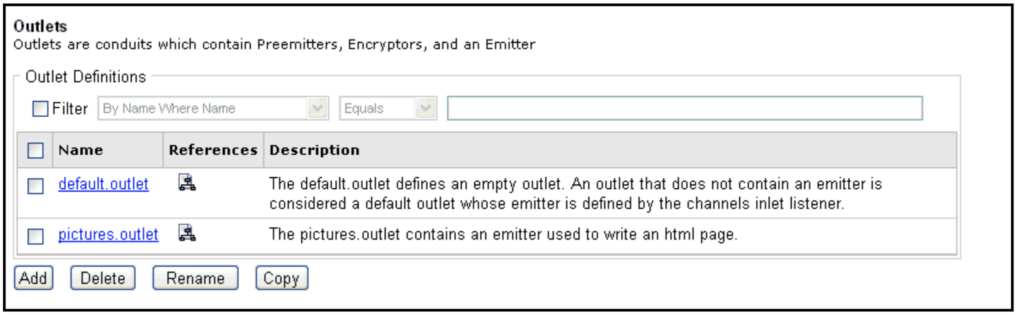
Procedure: How to Define an Outlet

To define an outlet using the iWay Service Manager Administration Console:



1. In the left console pane of the Registry menu, select *Outlets*.

   The Outlets pane opens.



Outlets
Outlets are conduits which contain Preemitters, Encryptors, and an Emitter

Outlet Definitions

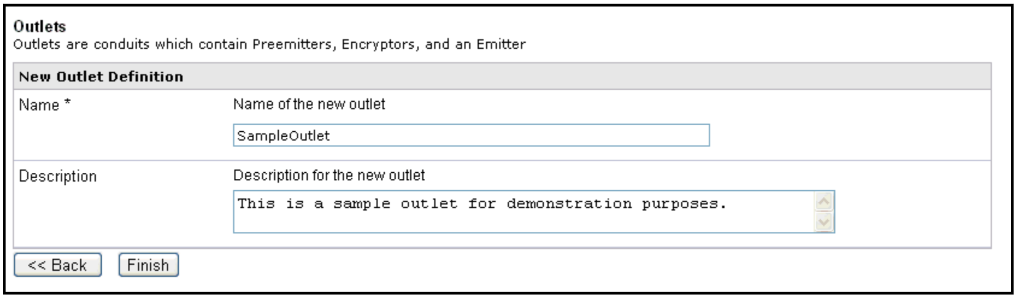
☐ Filter By Name Where Name | Equals

☐	Name	References	Description
☐	default.outlet	🖧	The default.outlet defines an empty outlet. An outlet that does not contain an emitter is considered a default outlet whose emitter is defined by the channels inlet listener.
☐	pictures.outlet	🖧	The pictures.outlet contains an emitter used to write an html page.

The table that is provided lists each outlet that is defined with a brief description. If you click the document schematic icon in the References column for a specific outlet, you will see which components are referencing that outlet.

2. Click *Add*.

   The New Outlet Definition pane opens.



Outlets
Outlets are conduits which contain Preemitters, Encryptors, and an Emitter

New Outlet Definition

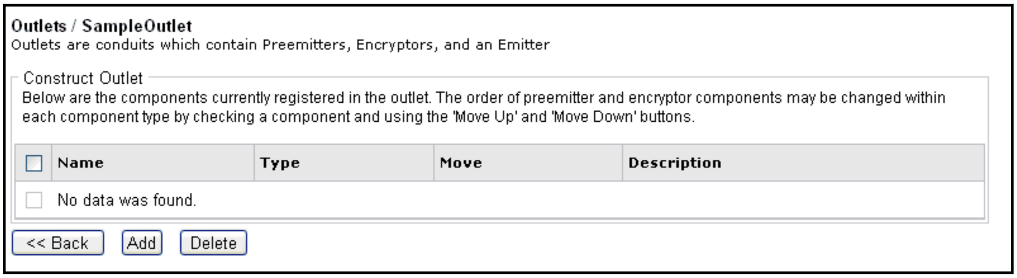
Name * — Name of the new outlet: SampleOutlet

Description — Description for the new outlet: This is a sample outlet for demonstration purposes.

3. Enter a name, for example, SampleOutlet, and description for the outlet.

4. Click *Finish*.

   The Construct Outlet pane opens.



Outlets / SampleOutlet
Outlets are conduits which contain Preemitters, Encryptors, and an Emitter

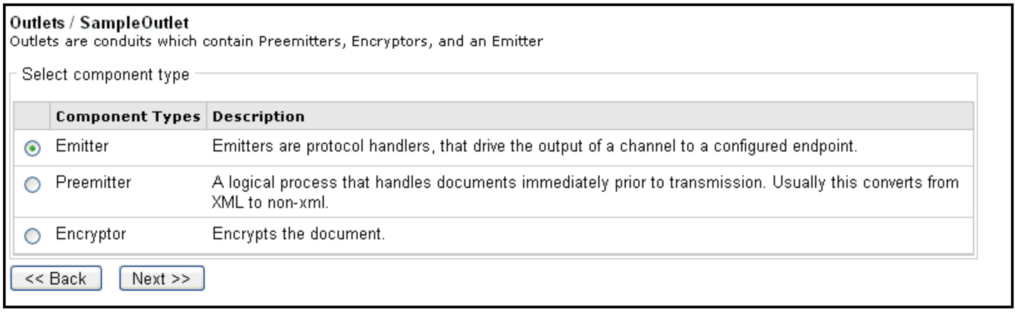
Construct Outlet
Below are the components currently registered in the outlet. The order of preemitter and encryptor components may be changed within each component type by checking a component and using the 'Move Up' and 'Move Down' buttons.

□	Name	Type	Move	Description
□	No data was found.			

   The table that is provided is used to list the components that are currently registered with the outlet.

5. Click *Add*.

   The Select component type pane opens.



Outlets / SampleOutlet
Outlets are conduits which contain Preemitters, Encryptors, and an Emitter

Select component type

	Component Types	Description
●	Emitter	Emitters are protocol handlers, that drive the output of a channel to a configured endpoint.
○	Preemitter	A logical process that handles documents immediately prior to transmission. Usually this converts from XML to non-xml.
○	Encryptor	Encrypts the document.

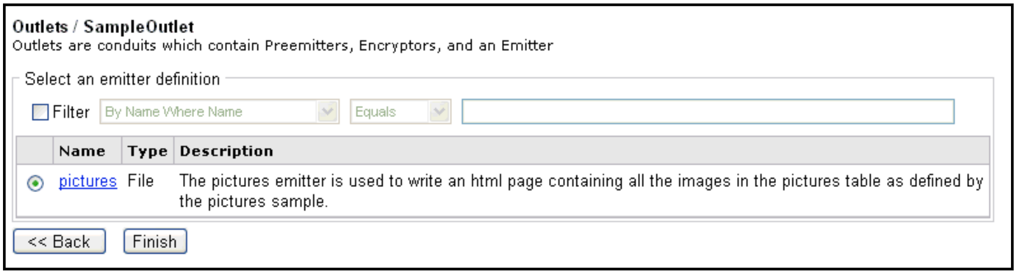
   The table that is provided lists the component types you can select and register with the outlet you are defining.

   - **Emitter** - Protocol handlers that send the output of a channel to a configured endpoint.
   - **Preemitter** - A logical process that handles documents immediately prior to transmission, for example, converting an XML document to non-XML.
   - **Encryptor** - Used to encrypt a document.

   An outlet that does not contain an emitter is considered a default outlet, whose emitter is defined by a channel’s inlet listener. A default outlet defines an empty outlet.

6. Select *Emitter* from the list of component types and click *Next*.

The Select an emitter definition pane opens.



Outlets / SampleOutlet

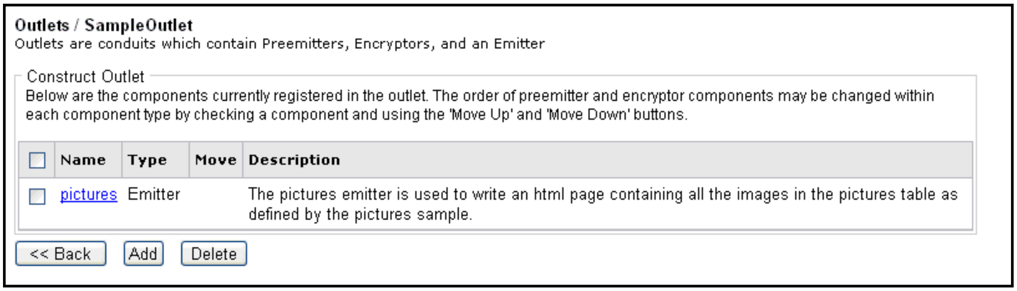
Outlets are conduits which contain Preemitters, Encryptors, and an Emitter

Select an emitter definition

	Name	Type	Description
●	pictures	File	The pictures emitter is used to write an html page containing all the images in the pictures table as defined by the pictures sample.

7. Select an available emitter, for example, pictures, from the list and click *Finish*.

   You are returned to the Construct Outlet pane, which now includes the emitter (pictures) you registered with your outlet (SampleOutlet).



Outlets / SampleOutlet

Outlets are conduits which contain Preemitters, Encryptors, and an Emitter

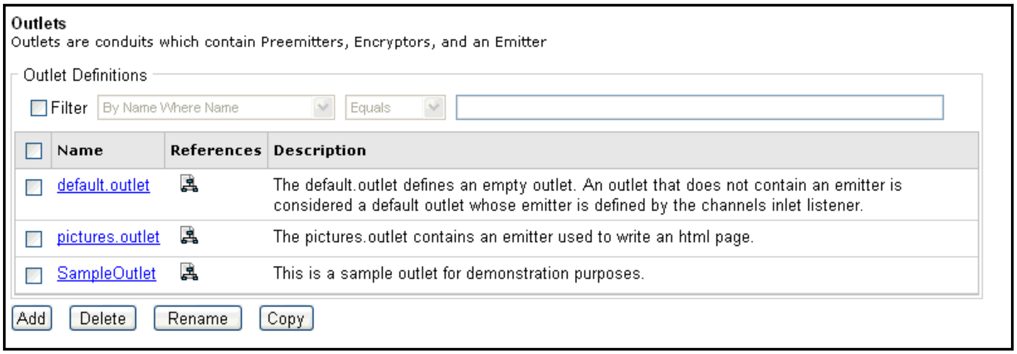
Construct Outlet

Below are the components currently registered in the outlet. The order of preemitter and encryptor components may be changed within each component type by checking a component and using the 'Move Up' and 'Move Down' buttons.

□	Name	Type	Move	Description
□	pictures	Emitter		The pictures emitter is used to write an html page containing all the images in the pictures table as defined by the pictures sample.

You can now add other components, such as preemitters or encryptors.

If you return to the main Outlets pane, you will notice that the outlet you just defined (SampleOutlet) has been added to the list, as shown in the following image.



Outlets

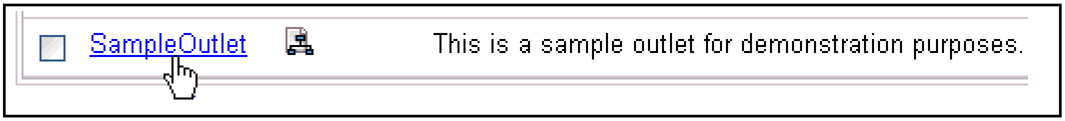
Outlets are conduits which contain Preemitters, Encryptors, and an Emitter

Outlet Definitions

□	Name	References	Description
□	default.outlet		The default.outlet defines an empty outlet. An outlet that does not contain an emitter is considered a default outlet whose emitter is defined by the channels inlet listener.
□	pictures.outlet		The pictures.outlet contains an emitter used to write an html page.
□	SampleOutlet		This is a sample outlet for demonstration purposes.

Procedure: How to Modify an Outlet

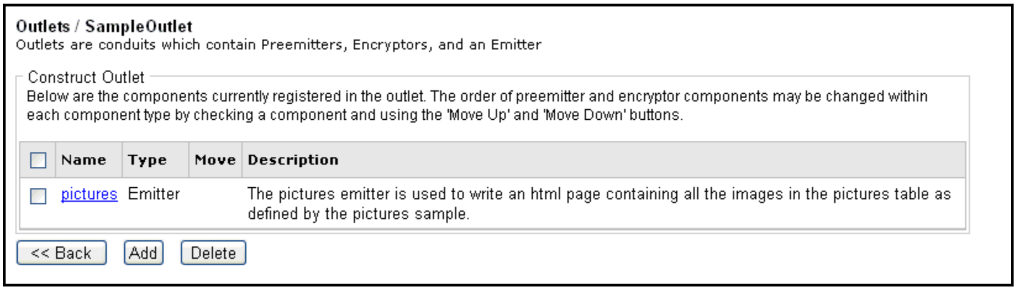
To modify an outlet:



□	SampleOutlet	This is a sample outlet for demonstration purposes.

1. Click the name of the outlet you wish to modify in the main Outlets pane, for example, SampleOutlet.

   The Construct Outlet pane opens.



Outlets / SampleOutlet
Outlets are conduits which contain Preemitters, Encryptors, and an Emitter

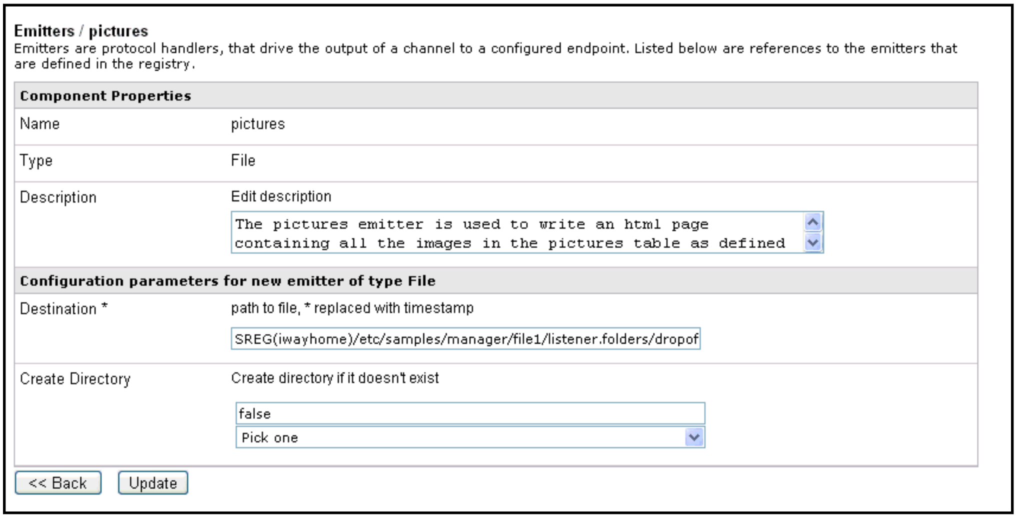
Construct Outlet
Below are the components currently registered in the outlet. The order of preemitter and encryptor components may be changed within each component type by checking a component and using the 'Move Up' and 'Move Down' buttons.

□	Name	Type	Move	Description
□	pictures	Emitter		The pictures emitter is used to write an html page containing all the images in the pictures table as defined by the pictures sample.

The table that is provided shows the component that has been registered for the outlet you are modifying. In this example, the pictures emitter is used.

2. Click the name of the emitter.

   The Emitters pane for the pictures emitter opens.



Emitters / pictures
Emitters are protocol handlers, that drive the output of a channel to a configured endpoint. Listed below are references to the emitters that are defined in the registry.

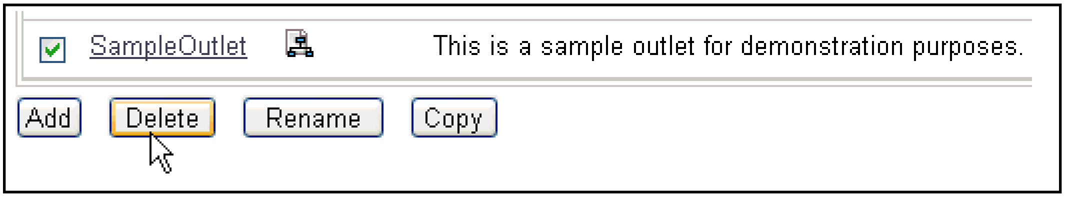
Component Properties	
Name	pictures
Type	File
Description	Edit description The pictures emitter is used to write an html page containing all the images in the pictures table as defined
Configuration parameters for new emitter of type File	
Destination *	path to file, * replaced with timestamp SREG(iwayhome)/etc/samples/manager/file1/listener.folders/dropof
Create Directory	Create directory if it doesn't exist false Pick one

3. Make your changes to the emitter properties as required and click *Update* when you are finished.

   The outlet is now modified.

Procedure: How to Delete an Outlet

To delete an outlet:



☑ SampleOutlet This is a sample outlet for demonstration purposes.

1. Click the check box next to the outlet you wish to delete, for example, SampleOutlet.
2. Click *Delete*.

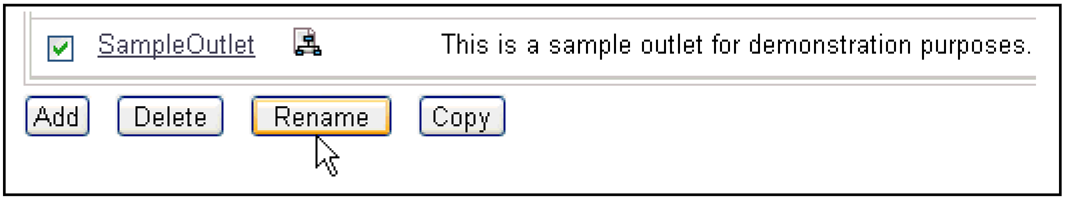
   A confirmation dialog box opens, prompting you to confirm the deletion of the outlet.
3. Click *OK*.

   The outlet is deleted from the system.

   Tip: To delete more than one outlet at once, you can select multiple check boxes in the Outlets pane and click *Delete*.

Procedure: How to Rename an Outlet

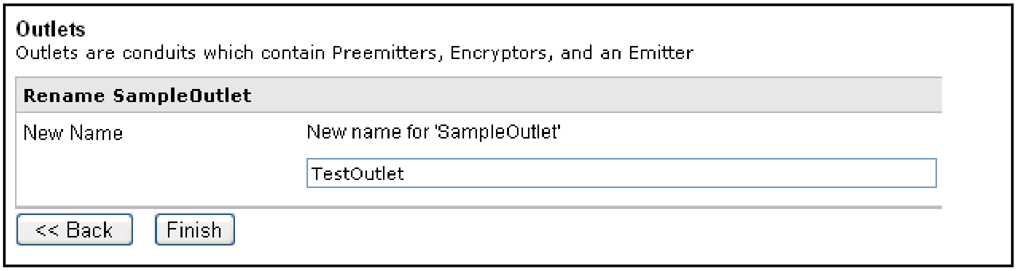
To rename an outlet:



☑ SampleOutlet This is a sample outlet for demonstration purposes.

1. Click the check box next to the outlet you wish to rename, for example, SampleOutlet.
2. Click *Rename*.

The Rename pane opens.



Outlets

Outlets are conduits which contain Preemitters, Encryptors, and an Emitter

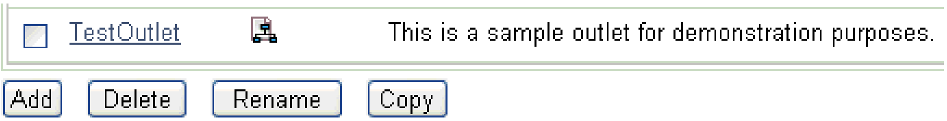
Rename SampleOutlet

New Name — New name for 'SampleOutlet'

TestOutlet

3. Type a name, for example, TestOutlet, in the New Name field and click *Finish*.

   You are returned to the main Outlets pane.

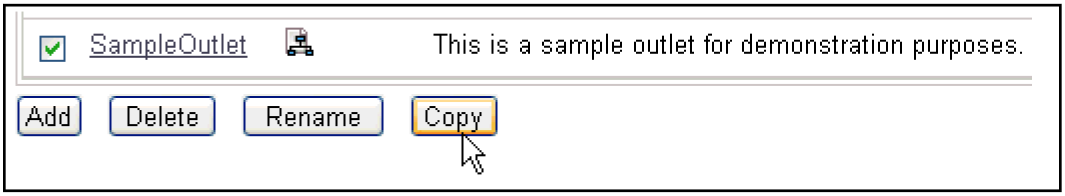


□ TestOutlet — This is a sample outlet for demonstration purposes.

The new name for the outlet you provided is shown.

Procedure: How to Copy an Outlet

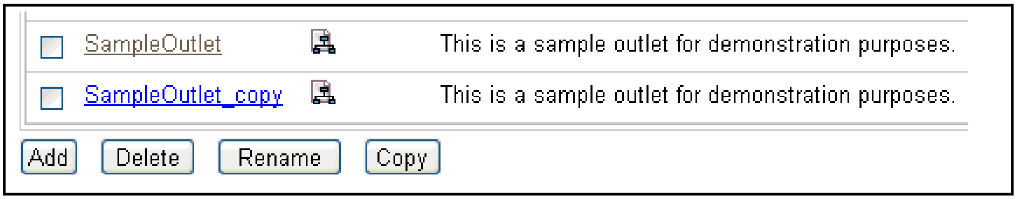
To copy an outlet:



☑ SampleOutlet — This is a sample outlet for demonstration purposes.

1. Click the check box next to the outlet you wish to copy, for example, SampleOutlet.
2. Click *Copy*.

   A copy of the outlet is made, as shown in the following image.



□ SampleOutlet — This is a sample outlet for demonstration purposes.

□ SampleOutlet_copy — This is a sample outlet for demonstration purposes.

Making copies of an outlet is useful for versioning and testing purposes.

Tip: To copy more than one outlet at once, you can select multiple check boxes in the Outlets pane and click *Copy*.

Outlet Strategies

iWay Service Manager provides mechanisms to support various routing strategies when using outlets. This section describes how you can use iSM to execute business logic and route documents to a particular location. The following topics are provided:

- Adding conditions to outlets.
- Configuring run-time options for outlets (On Success, On Error).

Procedure: How to Add Conditions to Outlets

To add conditions to outlets:

1. Open the channel you want to edit.
2. Click the *add conditions* icon in the outlet for which you want to set conditions.

   The Set Conditions pane appears.
3. Provide the condition, and then click *Back* to return to the channel pane.

   The icon changes to the edit icon to indicate that a condition has been set.

   If multiple outlets are defined for a channel and no conditions are added to the outlets, all of the outlets will be processed by iWay Service Manager.

   For more information on the conditions you can use, see the *iWay Service Manager Component and Functional Language Reference Guide*.

Procedure: How to Configure Outlet Run-Time Options

To configure outlet run-time options:

1. Open the channel you want to edit.
2. Click the *check* icon for the outlet you want to modify to set the run time option from On Success to On Error or the other way around, as appropriate.

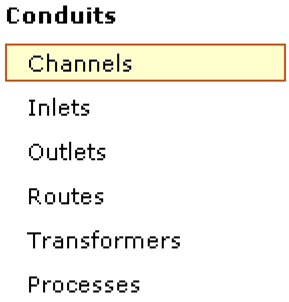
Constructing a Channel

After you have defined the necessary channel conduits (inlet, route, and outlet), you can combine these conduits and construct a channel using the iWay Service Manager Administration Console. Every channel is required to have an inlet, a route, and an outlet.

Note: In terms of case-sensitivity, the naming of channels and channel conduits (inlets, routes, and outlets) is platform dependent. On Windows, which is case-insensitive, defining channels and channel conduits using the same name (for example, TEST and test) is not permitted. However, on case-sensitive platforms, such as UNIX, this is permitted. For example, on UNIX, you can create two separate channels named TEST and test.

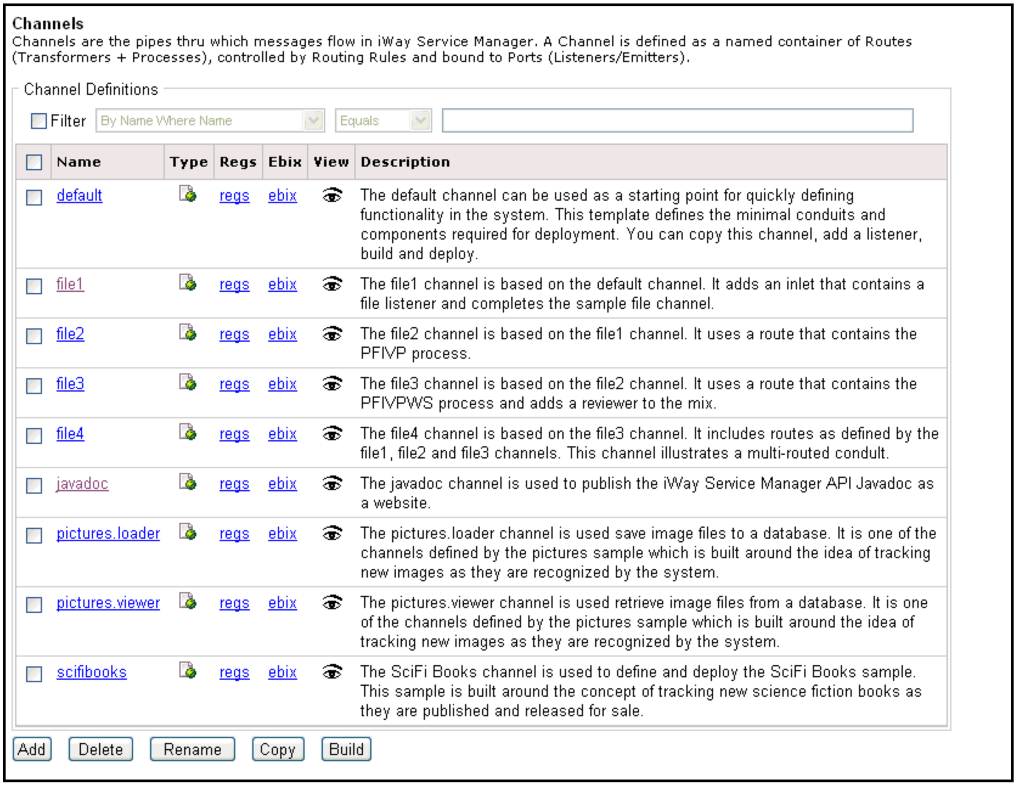
Procedure: How to Construct a Channel

To construct a channel using the iWay Service Manager Administration Console:



1. In the left console pane of the Registry menu, select *Channels*.

The Channels pane opens.



Channels

Channels are the pipes thru which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

Channel Definitions

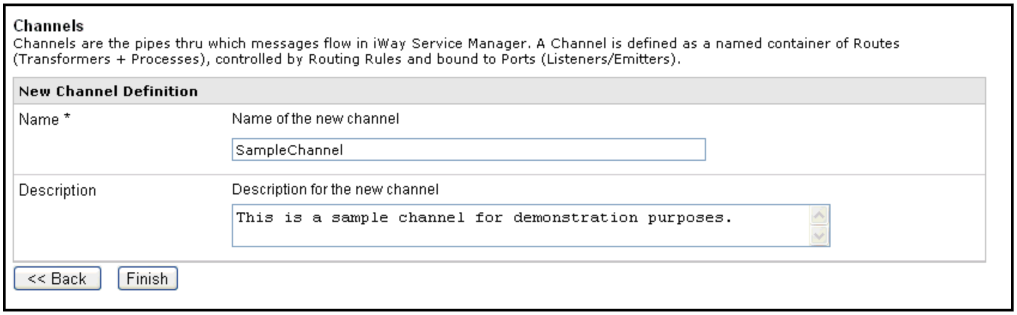
☐ Filter By Name Where Name Equals

☐	Name	Type	Regs	Ebix	View	Description
☐	default		regs	ebix	👁	The default channel can be used as a starting point for quickly defining functionality in the system. This template defines the minimal conduits and components required for deployment. You can copy this channel, add a listener, build and deploy.
☐	file1		regs	ebix	👁	The file1 channel is based on the default channel. It adds an inlet that contains a file listener and completes the sample file channel.
☐	file2		regs	ebix	👁	The file2 channel is based on the file1 channel. It uses a route that contains the PFIVP process.
☐	file3		regs	ebix	👁	The file3 channel is based on the file2 channel. It uses a route that contains the PFIVPWS process and adds a reviewer to the mix.
☐	file4		regs	ebix	👁	The file4 channel is based on the file3 channel. It includes routes as defined by the file1, file2 and file3 channels. This channel illustrates a multi-routed conduit.
☐	javadoc		regs	ebix	👁	The javadoc channel is used to publish the iWay Service Manager API Javadoc as a website.
☐	pictures.loader		regs	ebix	👁	The pictures.loader channel is used save image files to a database. It is one of the channels defined by the pictures sample which is built around the idea of tracking new images as they are recognized by the system.
☐	pictures.viewer		regs	ebix	👁	The pictures.viewer channel is used retrieve image files from a database. It is one of the channels defined by the pictures sample which is built around the idea of tracking new images as they are recognized by the system.
☐	scifibooks		regs	ebix	👁	The SciFi Books channel is used to define and deploy the SciFi Books sample. This sample is built around the concept of tracking new science fiction books as they are published and released for sale.

The table that is provided lists each channel that is defined with a brief description.

2. Click *Add*.

   The New Channel Definition pane opens.



Channels

Channels are the pipes thru which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

New Channel Definition

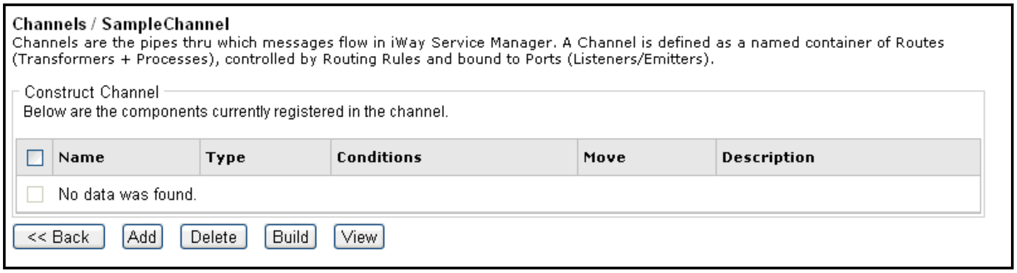
Name * — Name of the new channel: SampleChannel

Description — Description for the new channel: This is a sample channel for demonstration purposes.

3. Enter a name, for example, SampleChannel, and description for the channel.

4. Click *Finish*.

The Construct Channel pane opens.



Channels / SampleChannel
Channels are the pipes thru which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

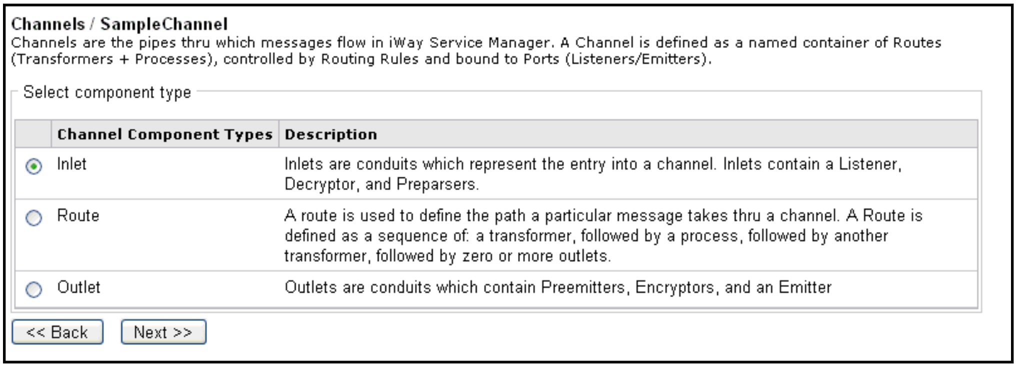
Construct Channel
Below are the components currently registered in the channel.

□	Name	Type	Conditions	Move	Description
□	No data was found.				

The table that is provided is used to list the components that are currently registered with the channel.

5. Click *Add*.

   The Select component type pane opens.



Channels / SampleChannel
Channels are the pipes thru which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

Select component type

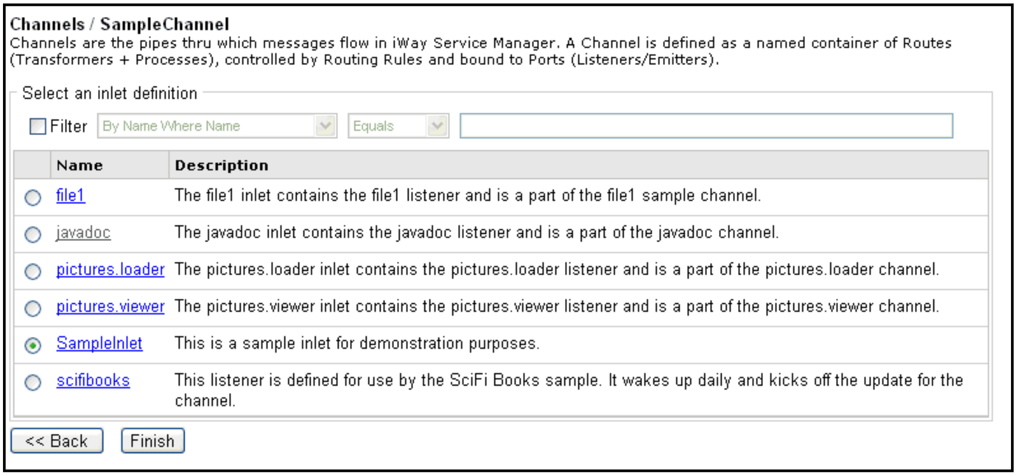
	Channel Component Types	Description
●	Inlet	Inlets are conduits which represent the entry into a channel. Inlets contain a Listener, Decryptor, and Preparsers.
○	Route	A route is used to define the path a particular message takes thru a channel. A Route is defined as a sequence of: a transformer, followed by a process, followed by another transformer, followed by zero or more outlets.
○	Outlet	Outlets are conduits which contain Preemitters, Encryptors, and an Emitter

The table that is provided lists the component types you can select and register with the channel you are defining.

- **Inlet** - Conduits that represent the entry into a channel.
- **Route** - Used to define the path a particular message takes through a channel.
- **Outlet** - Conduits that consist of Preemitters, Encryptors, and Emitters.

6. Select *Inlet* from the list of component types and click *Next*.

The select an inlet definition pane opens.



Channels / SampleChannel
Channels are the pipes thru which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

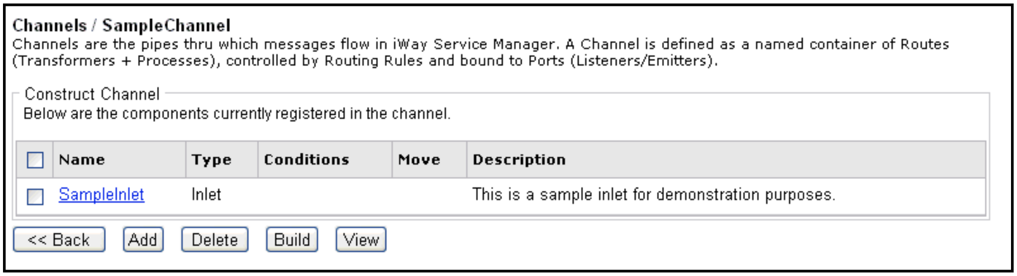
Select an inlet definition

□ Filter | By Name Where Name | Equals |

	Name	Description
○	file1	The file1 inlet contains the file1 listener and is a part of the file1 sample channel.
○	javadoc	The javadoc inlet contains the javadoc listener and is a part of the javadoc channel.
○	pictures.loader	The pictures.loader inlet contains the pictures.loader listener and is a part of the pictures.loader channel.
○	pictures.viewer	The pictures.viewer inlet contains the pictures.viewer listener and is a part of the pictures.viewer channel.
●	SampleInlet	This is a sample inlet for demonstration purposes.
○	scifibooks	This listener is defined for use by the SciFi Books sample. It wakes up daily and kicks off the update for the channel.

7. Select an available inlet, for example, SampleInlet, from the list and click *Finish*.

   You are returned to the Construct Channel pane, which now includes the inlet (SampleInlet) you defined earlier.



Channels / SampleChannel
Channels are the pipes thru which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

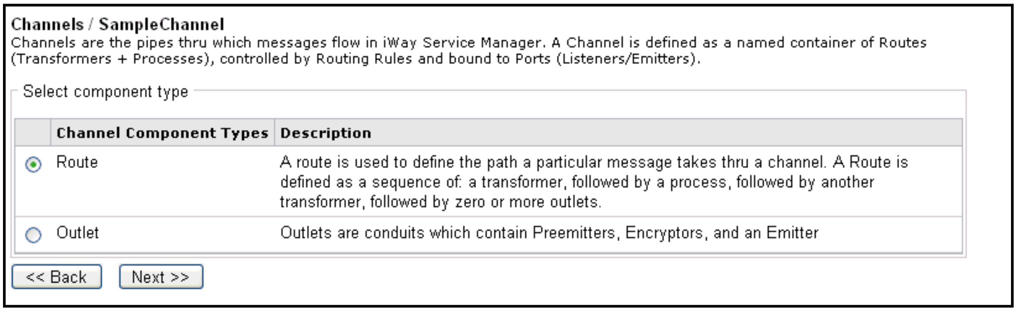
Construct Channel
Below are the components currently registered in the channel.

□	Name	Type	Conditions	Move	Description
□	SampleInlet	Inlet			This is a sample inlet for demonstration purposes.

   You are now ready to add a route to the channel.

8. Click *Add*.

   The Select component type pane opens.



Channels / SampleChannel
Channels are the pipes thru which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

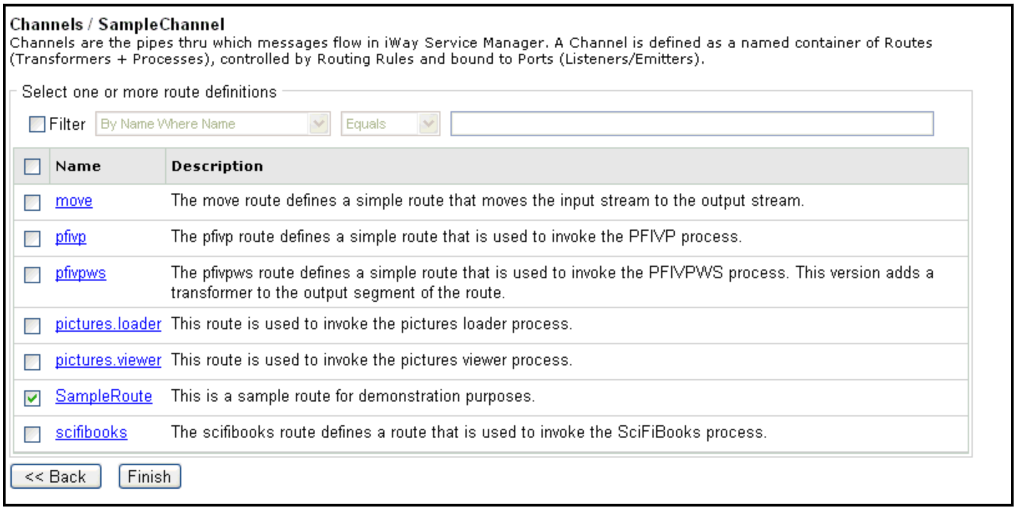
Select component type

	Channel Component Types	Description
●	Route	A route is used to define the path a particular message takes thru a channel. A Route is defined as a sequence of: a transformer, followed by a process, followed by another transformer, followed by zero or more outlets.
○	Outlet	Outlets are conduits which contain Preemitters, Encryptors, and an Emitter

Notice that only the Route and Outlet component types are listed, since you have already added an inlet to the channel.

9. Select *Route* from the list of component types and click *Next*.

   The select one or more route definitions pane opens.



Channels / SampleChannel
Channels are the pipes thru which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

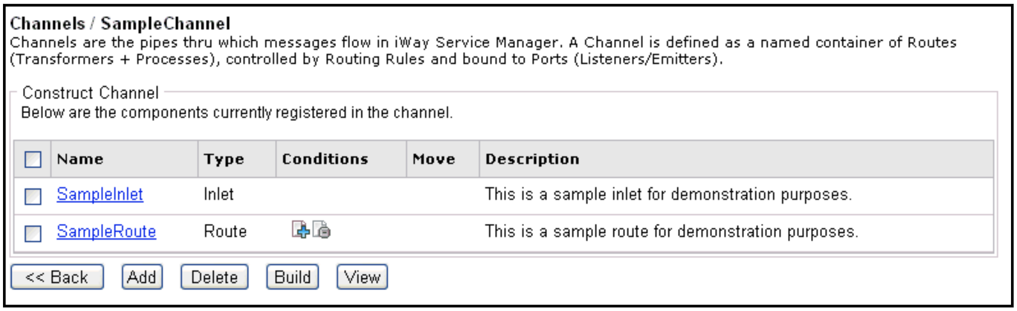
Select one or more route definitions

□ Filter By Name Where Name | Equals

□	Name	Description
□	move	The move route defines a simple route that moves the input stream to the output stream.
□	pfivp	The pfivp route defines a simple route that is used to invoke the PFIVP process.
□	pfivpws	The pfivpws route defines a simple route that is used to invoke the PFIVPWS process. This version adds a transformer to the output segment of the route.
□	pictures.loader	This route is used to invoke the pictures loader process.
□	pictures.viewer	This route is used to invoke the pictures viewer process.
☑	SampleRoute	This is a sample route for demonstration purposes.
□	scifibooks	The scifibooks route defines a route that is used to invoke the SciFiBooks process.

10. Select an available route, for example, SampleRoute, from the list and click *Finish*.

    You are returned to the Construct Channel pane, which now includes the inlet (SampleInlet) and route (SampleRoute) you defined earlier.



Channels / SampleChannel
Channels are the pipes thru which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

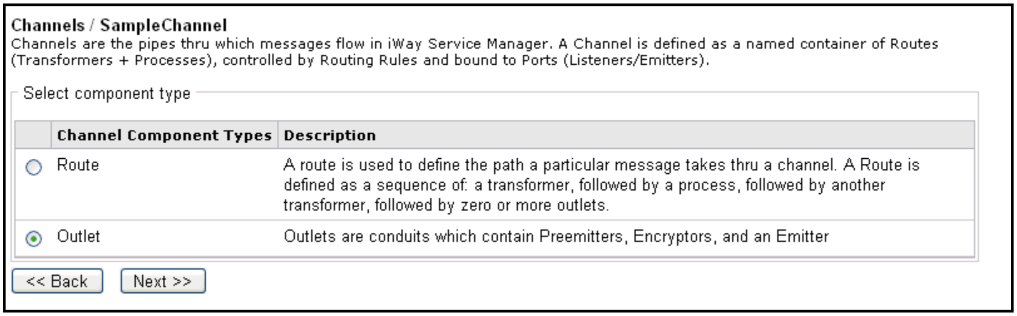
Construct Channel
Below are the components currently registered in the channel.

□	Name	Type	Conditions	Move	Description
□	SampleInlet	Inlet			This is a sample inlet for demonstration purposes.
□	SampleRoute	Route			This is a sample route for demonstration purposes.

    You are now ready to add an outlet to the channel.

11. Click *Add*.

The Select component type pane opens.



Channels / SampleChannel

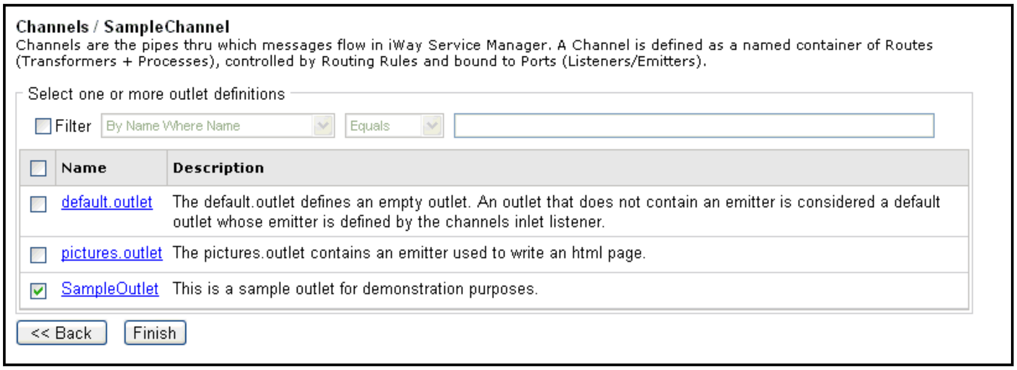
Channels are the pipes thru which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

Select component type

	Channel Component Types	Description
○	Route	A route is used to define the path a particular message takes thru a channel. A Route is defined as a sequence of: a transformer, followed by a process, followed by another transformer, followed by zero or more outlets.
●	Outlet	Outlets are conduits which contain Preemitters, Encryptors, and an Emitter

12. Select *Outlet* from the list of component types and click *Next*.

    The select one or more outlet definitions pane opens.



Channels / SampleChannel

Channels are the pipes thru which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

Select one or more outlet definitions

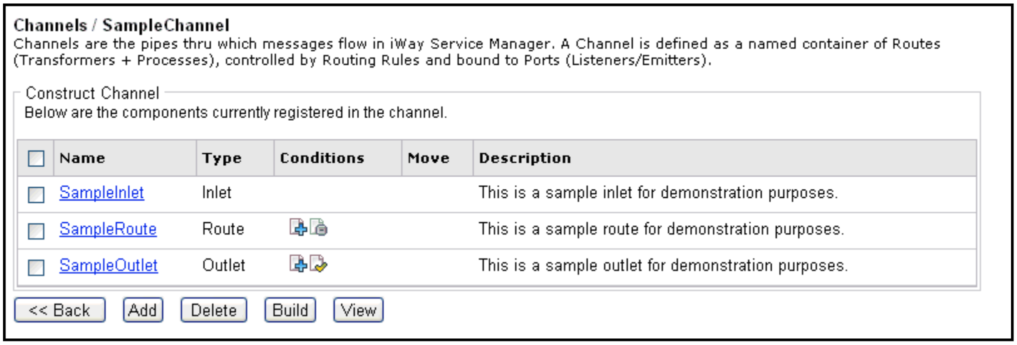
□ Filter By Name Where Name Equals

□	Name	Description
□	default.outlet	The default.outlet defines an empty outlet. An outlet that does not contain an emitter is considered a default outlet whose emitter is defined by the channels inlet listener.
□	pictures.outlet	The pictures.outlet contains an emitter used to write an html page.
☑	SampleOutlet	This is a sample outlet for demonstration purposes.

13. Select an available outlet, for example, SampleOutlet, from the list and click *Finish*.

    You can assign multiple routes and multiple outlets to the channel.

    You are returned to the Construct Channel pane, which now includes the inlet (SampleInlet), route (SampleRoute), and outlet (SampleOutlet) you defined earlier.



Channels / SampleChannel

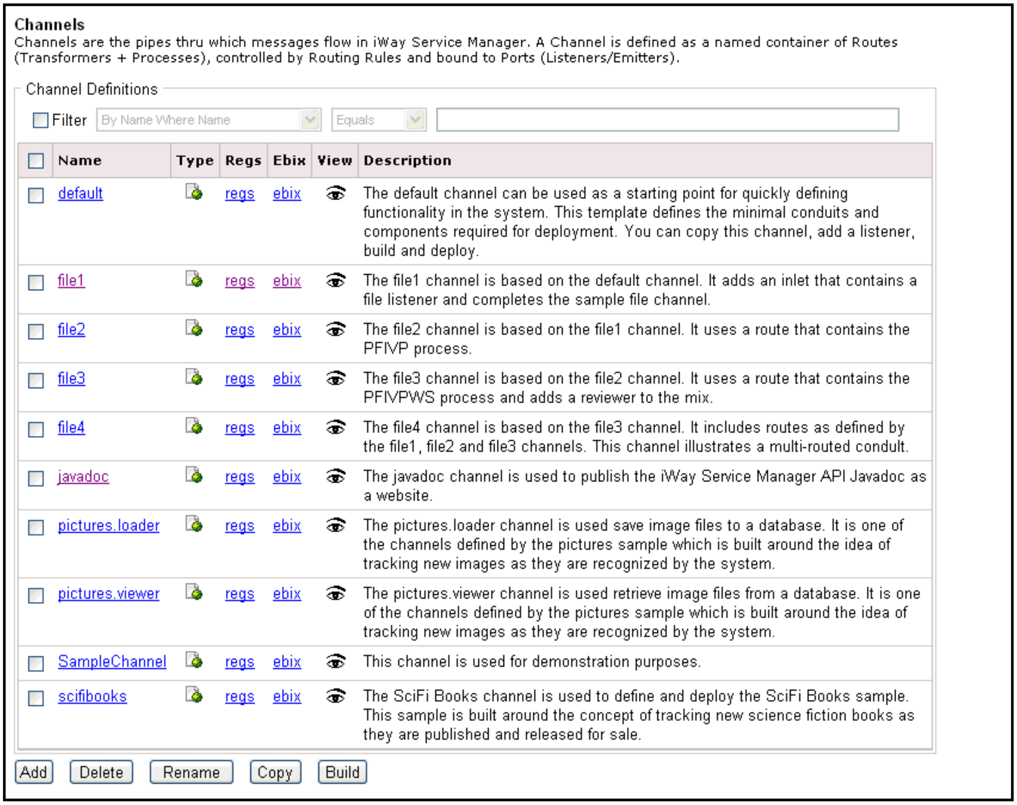
Channels are the pipes thru which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

Construct Channel

Below are the components currently registered in the channel.

□	Name	Type	Conditions	Move	Description
□	SampleInlet	Inlet			This is a sample inlet for demonstration purposes.
□	SampleRoute	Route			This is a sample route for demonstration purposes.
□	SampleOutlet	Outlet			This is a sample outlet for demonstration purposes.

If you return to the main channels pane, you will notice that the channel you just constructed (SampleChannel) has been added to the list, as shown in the following image.



Channels

Channels are the pipes thru which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

Channel Definitions

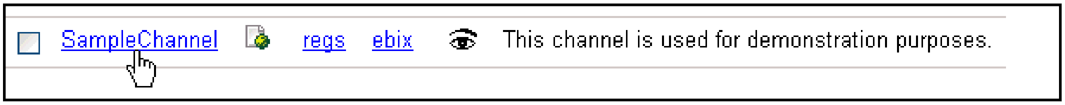
□ Filter | By Name Where Name | Equals |

□	Name	Type	Regs	Ebix	View	Description
□	default	🗎	regs	ebix	👁	The default channel can be used as a starting point for quickly defining functionality in the system. This template defines the minimal conduits and components required for deployment. You can copy this channel, add a listener, build and deploy.
□	file1	🗎	regs	ebix	👁	The file1 channel is based on the default channel. It adds an inlet that contains a file listener and completes the sample file channel.
□	file2	🗎	regs	ebix	👁	The file2 channel is based on the file1 channel. It uses a route that contains the PFIVP process.
□	file3	🗎	regs	ebix	👁	The file3 channel is based on the file2 channel. It uses a route that contains the PFIVPWS process and adds a reviewer to the mix.
□	file4	🗎	regs	ebix	👁	The file4 channel is based on the file3 channel. It includes routes as defined by the file1, file2 and file3 channels. This channel illustrates a multi-routed conduit.
□	javadoc	🗎	regs	ebix	👁	The javadoc channel is used to publish the iWay Service Manager API Javadoc as a website.
□	pictures.loader	🗎	regs	ebix	👁	The pictures.loader channel is used save image files to a database. It is one of the channels defined by the pictures sample which is built around the idea of tracking new images as they are recognized by the system.
□	pictures.viewer	🗎	regs	ebix	👁	The pictures.viewer channel is used retrieve image files from a database. It is one of the channels defined by the pictures sample which is built around the idea of tracking new images as they are recognized by the system.
□	SampleChannel	🗎	regs	ebix	👁	This channel is used for demonstration purposes.
□	scifibooks	🗎	regs	ebix	👁	The SciFi Books channel is used to define and deploy the SciFi Books sample. This sample is built around the concept of tracking new science fiction books as they are published and released for sale.

After you have designed your channel, you are ready to build and deploy it into a run-time environment. For more information on building, deploying, and repairing channels, see *Operations and Monitoring* on page 389.

Procedure: How to Modify a Channel

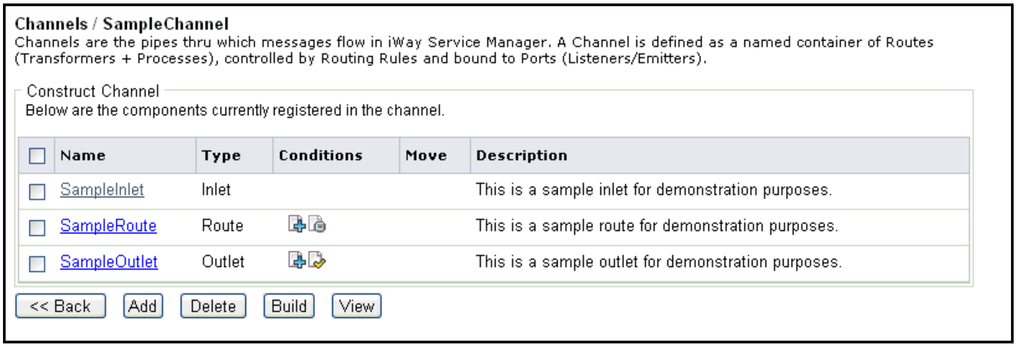
To modify a channel:



□	SampleChannel	🗎	regs	ebix	👁	This channel is used for demonstration purposes.

1. Click the channel name you wish to modify in the main Channels pane.

The Construct Channel pane opens.



Channels / SampleChannel
Channels are the pipes thru which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

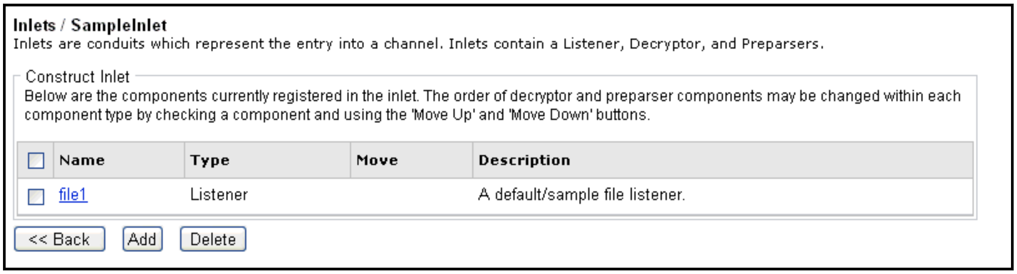
Construct Channel
Below are the components currently registered in the channel.

□	Name	Type	Conditions	Move	Description
□	SampleInlet	Inlet			This is a sample inlet for demonstration purposes.
□	SampleRoute	Route			This is a sample route for demonstration purposes.
□	SampleOutlet	Outlet			This is a sample outlet for demonstration purposes.

The table that is provided lists all the channel components (inlet, routes, and outlets) that have been defined for that channel.

2. Click the name of the component, for example, SampleInlet.

   The Construct Inlet pane opens.



Inlets / SampleInlet
Inlets are conduits which represent the entry into a channel. Inlets contain a Listener, Decryptor, and Preparsers.

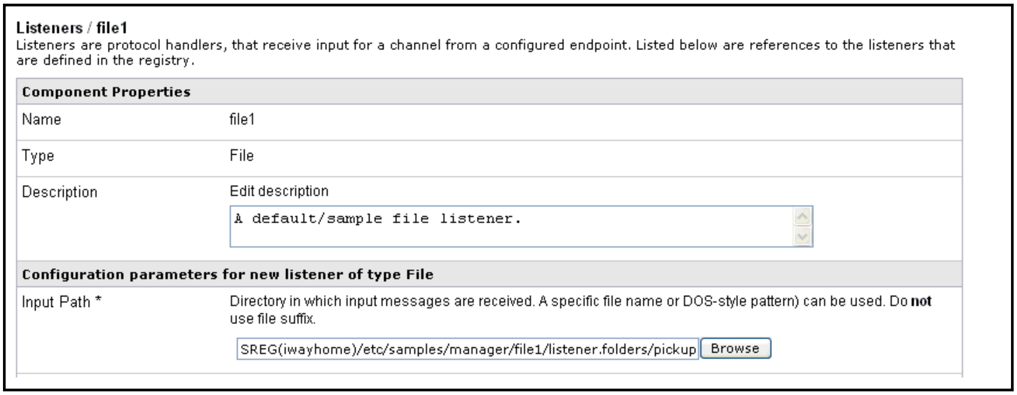
Construct Inlet
Below are the components currently registered in the inlet. The order of decryptor and preparser components may be changed within each component type by checking a component and using the 'Move Up' and 'Move Down' buttons.

□	Name	Type	Move	Description
□	file1	Listener		A default/sample file listener.

The table that is provided shows the component that has been registered for the inlet. In this example, the file1 listener is used.

3. Click the name of the listener.

The Listeners pane for the file1 listener opens.



Listeners / **file1**

Listeners are protocol handlers, that receive input for a channel from a configured endpoint. Listed below are references to the listeners that are defined in the registry.

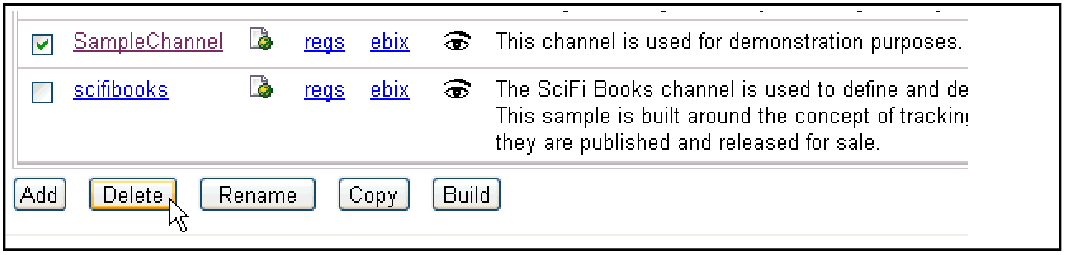
Component Properties	
Name	file1
Type	File
Description	Edit description A default/sample file listener.
Configuration parameters for new listener of type File	
Input Path *	Directory in which input messages are received. A specific file name or DOS-style pattern) can be used. Do **not** use file suffix. SREG(iwayhome)/etc/samples/manager/file1/listener.folders/pickup

4. Make your changes to the listener properties as required and click *Update* when you are finished.

   The channel is now modified.

Procedure: How to Delete a Channel

To delete a channel:



☑ SampleChannel	regs	ebix		This channel is used for demonstration purposes.
☐ scifibooks	regs	ebix		The SciFi Books channel is used to define and de This sample is built around the concept of trackin they are published and released for sale.

1. Click the check box next to the channel you wish to delete, for example, SampleChannel.
2. Click *Delete*.

   A confirmation dialog box opens, prompting you to confirm the deletion of the channel.
3. Click *OK*.

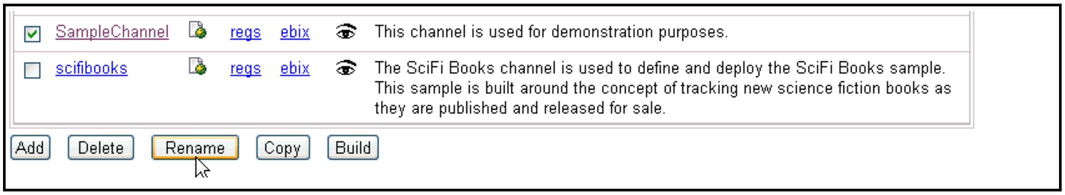
   The channel is deleted from the system.

   Tip: To delete more than one channel at once, you can select multiple check boxes in the Channels pane and click *Delete*.

   Note: Deleting a channel from the registry does not have any impact on deployed channels.

Procedure: How to Rename a Channel

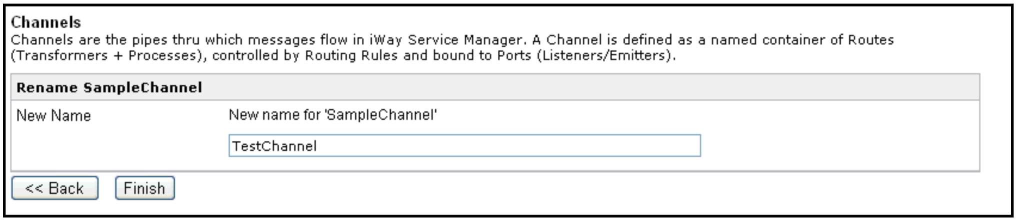
To rename a channel:



☑	SampleChannel		regs	ebix		This channel is used for demonstration purposes.
□	scifibooks		regs	ebix		The SciFi Books channel is used to define and deploy the SciFi Books sample. This sample is built around the concept of tracking new science fiction books as they are published and released for sale.

1. Click the check box next to the channel you wish to rename, for example, SampleChannel.
2. Click *Rename*.

   The Rename pane opens.



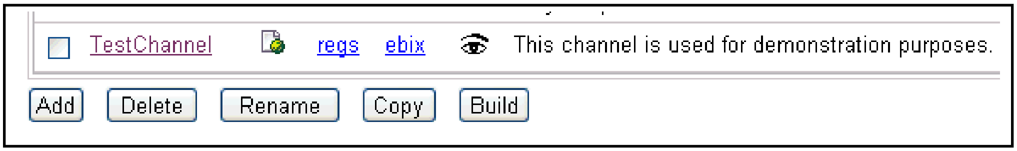
Channels
Channels are the pipes thru which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

Rename SampleChannel

New Name — New name for 'SampleChannel': TestChannel

3. Type a name, for example, TestChannel, in the New Name field and click *Finish*.

   You are returned to the main Channels pane.

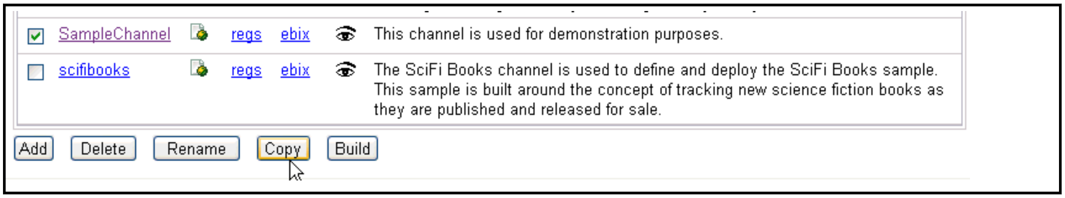


□	TestChannel		regs	ebix		This channel is used for demonstration purposes.

The new name for the channel you provided is shown. If this channel has already been deployed into production, the prior name and version will remain in production until it is undeployed (see *How to Undeploy a Channel* on page 442).

Procedure: How to Copy a Channel

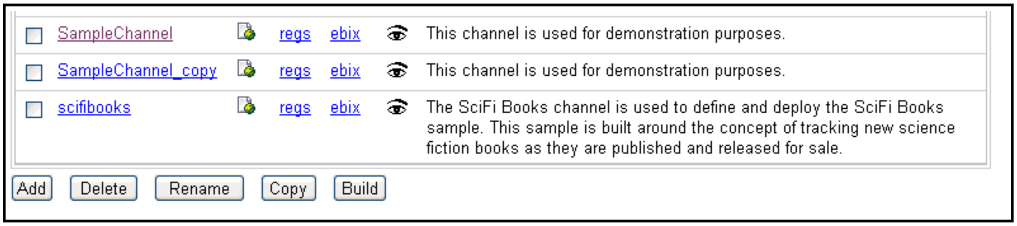
To copy a channel:



☑	SampleChannel	regs	ebix	This channel is used for demonstration purposes.
□	scifibooks	regs	ebix	The SciFi Books channel is used to define and deploy the SciFi Books sample. This sample is built around the concept of tracking new science fiction books as they are published and released for sale.

1. Click the check box next to the channel you wish to copy, for example, SampleChannel.
2. Click *Copy*.

   A copy of the channel is made, as shown in the following image.



□	SampleChannel	regs	ebix	This channel is used for demonstration purposes.
□	SampleChannel_copy	regs	ebix	This channel is used for demonstration purposes.
□	scifibooks	regs	ebix	The SciFi Books channel is used to define and deploy the SciFi Books sample. This sample is built around the concept of tracking new science fiction books as they are published and released for sale.

Making copies of a channel is useful for versioning and testing purposes.

Tip: To copy more than one channel at once, you can select multiple check boxes in the Channels pane and click *Copy*.

Specifying Fixed or Dynamic Routing

After you have constructed a channel in the iWay Service Manager Administration Console, you can specify whether fixed or dynamic routing should be used during channel processing.

Dynamic routing is a routing method for a channel that reevaluates the available routes at each step of the messaging process. As a result, the routing can respond to changes in the message and execution environment. By default, every channel that is created is automatically set to use dynamic routing.

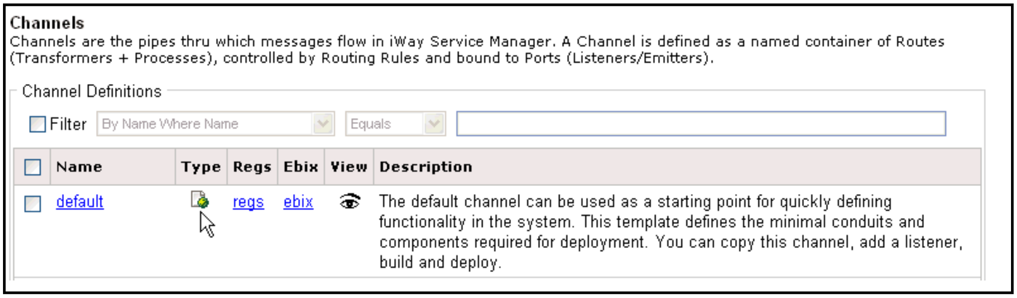
Fixed routing is a routing method for a channel in which a selected route in a channel remains in force for the duration of the messaging process.

Procedure: How to Specify Fixed Routing for a Channel

To specify fixed routing:

1. In the left console pane of the Registry menu, select *Channels*.

The Channels pane opens.



Channels
Channels are the pipes thru which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

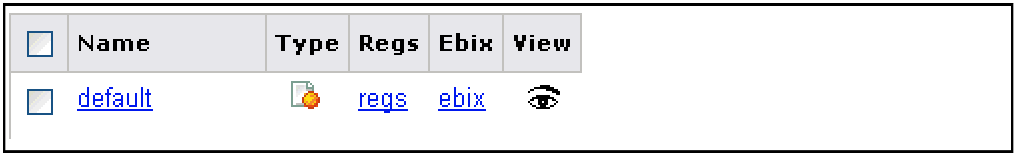
Channel Definitions

□ Filter By Name Where Name | Equals

□	Name	Type	Regs	Ebix	View	Description
□	default		regs	ebix	👁	The default channel can be used as a starting point for quickly defining functionality in the system. This template defines the minimal conduits and components required for deployment. You can copy this channel, add a listener, build and deploy.

2. Click the *route* icon in the Type column.

   The route icon in the Type column changes color (to orange) to indicate that the route is now fixed, as shown in the following image.



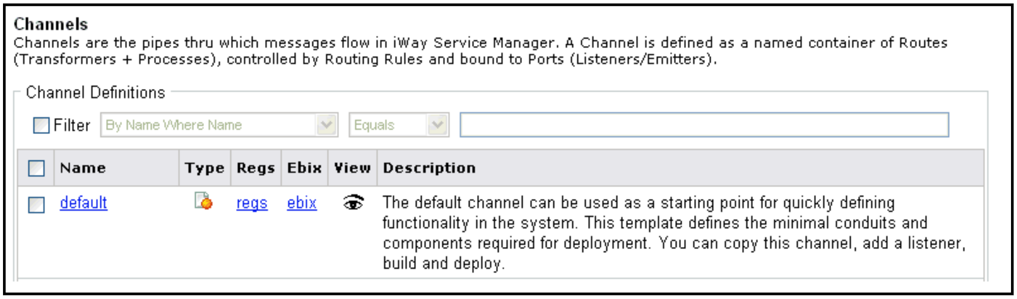
□	Name	Type	Regs	Ebix	View
□	default		regs	ebix	👁

Procedure: How to Specify Dynamic Routing for a Channel

To specify dynamic routing:

1. In the left console pane of the Registry menu, select *Channels*.

   The Channels pane opens.



Channels
Channels are the pipes thru which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

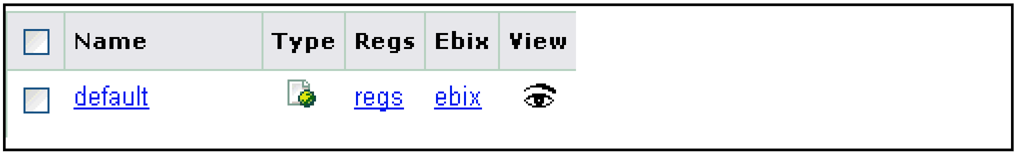
Channel Definitions

□ Filter By Name Where Name | Equals

□	Name	Type	Regs	Ebix	View	Description
□	default		regs	ebix	👁	The default channel can be used as a starting point for quickly defining functionality in the system. This template defines the minimal conduits and components required for deployment. You can copy this channel, add a listener, build and deploy.

2. Click the *route* icon in the Type column that is currently set to fixed routing.

The route icon in the Type column changes color (to green) to indicate that the route is now dynamic, as shown in the following image.



□	Name	Type	Regs	Ebix	View
□	default	📄	regs	ebix	👁

Adding Register Sets

In iWay Service Manager, a Special Register (SREG) is a name-value pair that defines a variable that is carried throughout the system. Once defined, this variable is available to be bound to all components of the system. For example, a document arriving on HTTP can report its origin IP address in the special register IP. Using the expression _sreg(IP) in the channel directs the output onto a queue with the name of the IP address. For a complete list of special registers provided, see the *iWay Service Manager Programmer's Guide*. For more information on defining a special register of your own, see *Configuring General Properties Using the Console* on page 111.

A common use of special register routing is to send the contents of a document to the appropriate database instance configured for the internationalization encoding of the document. You can use all of the iWay-supplied special registers to route a document dynamically.

Procedure: How to Add a Register Set

To add a register set to a channel:

1. In the left console pane of the Registry menu, select *Channels*.

   The Channels pane opens.

2. Click *Regs* for the channel you want to modify.

   The Assign register pane opens.

3. Select a register and click *Finish*.

4. Click *Back* to return to the Channels pane.
   If a register set is added to a channel, the channel must be rebuilt and redeployed for new values to take effect.

Adding Ebix Components

Ebix is a collection of metadata archives that define the structure of the data. iWay Software provides various Ebix files used in conjunction with the iWay Format Adapters.

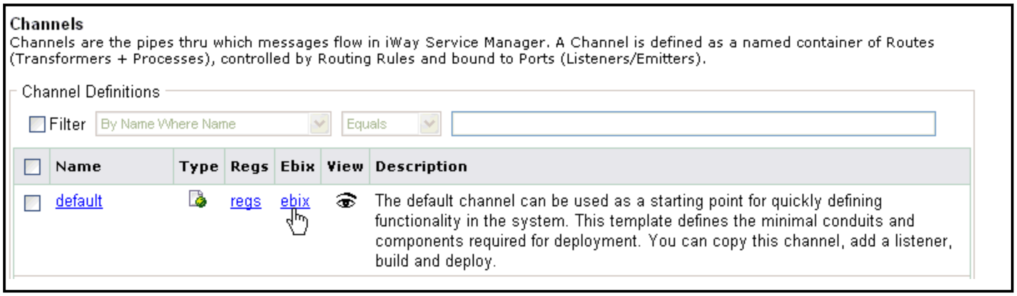
Note: Ebix must be defined in the register before they can be added to a channel. If required, you can add multiple ebix components to a channel.

Procedure: How to Add an Ebix Component to a Channel

To add an ebix component to a channel:

1. In the left console pane of the Registry menu, select *Channels*.

   The Channels pane opens.



Channels

Channels are the pipes thru which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

Channel Definitions

□ Filter | By Name Where Name | Equals |

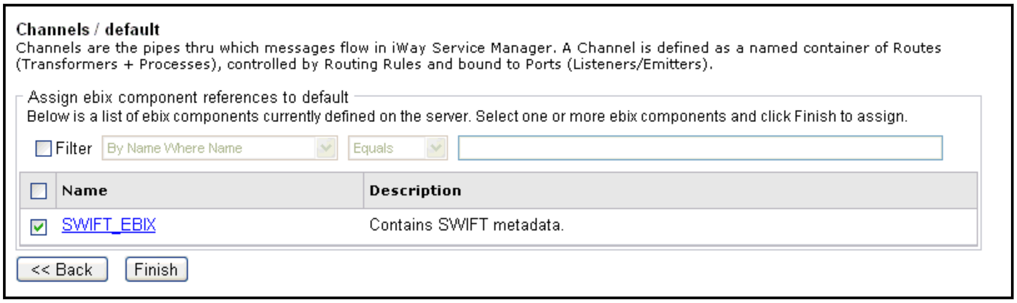
□	Name	Type	Regs	Ebix	View	Description
□	default		regs	ebix		The default channel can be used as a starting point for quickly defining functionality in the system. This template defines the minimal conduits and components required for deployment. You can copy this channel, add a listener, build and deploy.

2. Click *ebix* in the Ebix column.

   The Add ebix components pane opens.

3. Click *Add*.

   The Assign ebix component references pane opens.



Channels / default

Channels are the pipes thru which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

Assign ebix component references to default

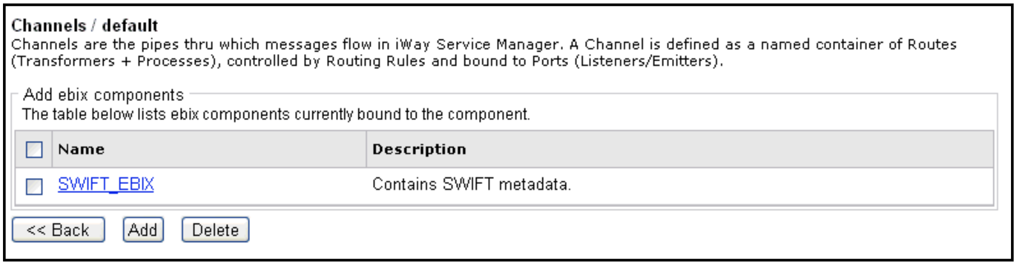
Below is a list of ebix components currently defined on the server. Select one or more ebix components and click Finish to assign.

□ Filter | By Name Where Name | Equals |

□	Name	Description
☑	SWIFT_EBIX	Contains SWIFT metadata.

4. Select the check box next to the ebix, for example, SWIFT_EBIX, you want to add and click *Finish*.

You are returned to the Add ebix components pane.



Channels / default

Channels are the pipes thru which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

Add ebix components

The table below lists ebix components currently bound to the component.

□	Name	Description
□	SWIFT_EBIX	Contains SWIFT metadata.

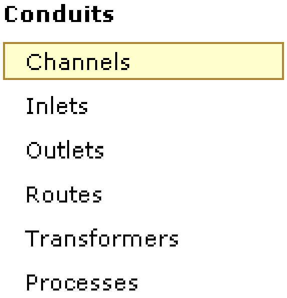
The ebix you added is now assigned to your channel.

Building a Channel

After restructuring a channel, building a channel is the first stage in channel management. This process compiles all the registered channel components (inlet, route, and outlet) and validates the combination of components you have selected. Building a channel makes it available to deploying to one or more managed servers

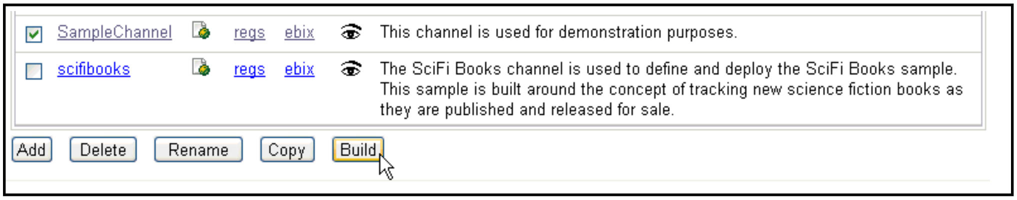
Procedure: How to Build a Channel

To build a channel:



1. In the left console pane of the Registry menu, select *Channels*.

   The Channels pane opens.

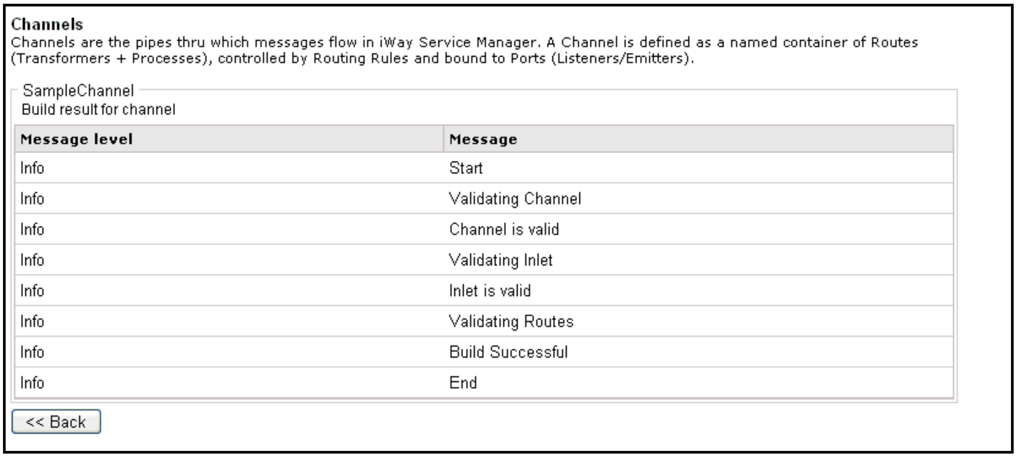


☑	SampleChannel		regs	ebix		This channel is used for demonstration purposes.
□	scifibooks		regs	ebix		The SciFi Books channel is used to define and deploy the SciFi Books sample. This sample is built around the concept of tracking new science fiction books as they are published and released for sale.

The table that is provided lists each channel that is defined with a brief description.

2. Select the check box next to the channel you want to build, for example, SampleChannel, and click *Build*.

   The build result pane for the channel opens.



Channels

Channels are the pipes thru which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

SampleChannel

Build result for channel

Message level	Message
Info	Start
Info	Validating Channel
Info	Channel is valid
Info	Validating Inlet
Info	Inlet is valid
Info	Validating Routes
Info	Build Successful
Info	End

   Each validation step is listed in the table and includes the final build result. If no errors are listed, you have successfully built a channel, which is now ready to be deployed.

   Tip: To build more than one channel at once, you can select multiple check boxes in the Channels pane and click *Build*.

3. Click *Back* to return to the Channels pane.

Chapter 9

Using iWay Service Manager Tools

This section describes how to use the tools and applications that are available under the Tools menu in the iWay Service Manager Administration Console.

In this chapter:

Using the Log Viewer

The iWay Service Manager Administration Console contains a powerful, multi-level logging facility for problem determination and remediation. The Log Viewer manages the display properties of system debugging information when the logging and tracing functions are activated. It filters and displays debugging information as each transaction is received and processed. The Log Viewer also displays the date/time range, type, source, and message of every trace entry.

For more information on using the Log Viewer, see *Diagnostics, Tracing, and Logging* on page 471.

Using the Package Manager

Packages are specially designed files that contain components, metadata, and configuration information and can be used to move runtime components such as SREGS and providers. Packages can be added or removed from a specific server instance. This section describes how to use the Package Manager to manage the addition and deletion of functionality within an iWay Service Manager run-time configuration.

Packages are archive files that contain components, metadata and configuration information. Packages can be are installed/uninstalled to apply their contents to a specific server instance.

Note: Creating archives instead of packages is recommended when migrating between different environments (for example, development, test, and production). For more information on how to create an archive, see *Using the Archive Manager* on page 365.

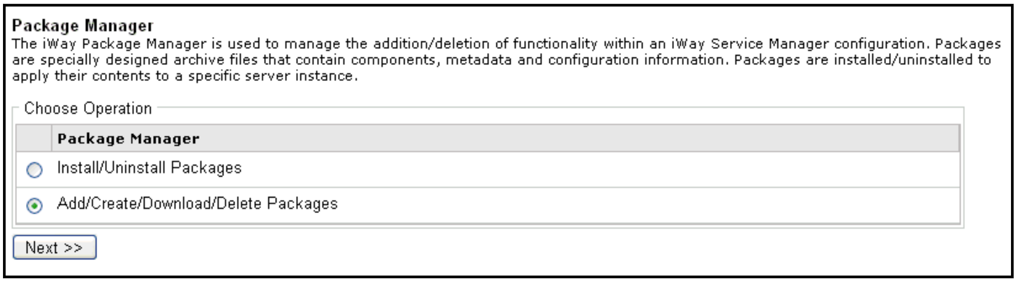
Adding a Package

In order to install a package, it must be in a predefined location within the server machine's file system. Add will take a package zip file from any location accessible by your browser and copy (upload) it to the appropriate directory on the server machine.

Procedure: How to Add a Package

1. Select the *Add/Create/Download/Delete Packages* button.

   The following image shows the Package Manager - Choose Operation page with the Add/Create/Download/Delete Packages button selected.



Package Manager

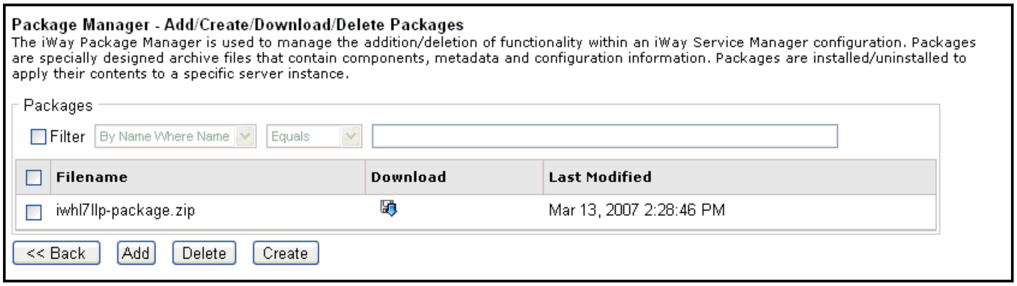
The iWay Package Manager is used to manage the addition/deletion of functionality within an iWay Service Manager configuration. Packages are specially designed archive files that contain components, metadata and configuration information. Packages are installed/uninstalled to apply their contents to a specific server instance.

Choose Operation

	Package Manager
○	Install/Uninstall Packages
●	Add/Create/Download/Delete Packages

2. Click *Next*.

   A list of already uploaded Package files is displayed, as shown in the following image.



Package Manager - Add/Create/Download/Delete Packages

The iWay Package Manager is used to manage the addition/deletion of functionality within an iWay Service Manager configuration. Packages are specially designed archive files that contain components, metadata and configuration information. Packages are installed/uninstalled to apply their contents to a specific server instance.

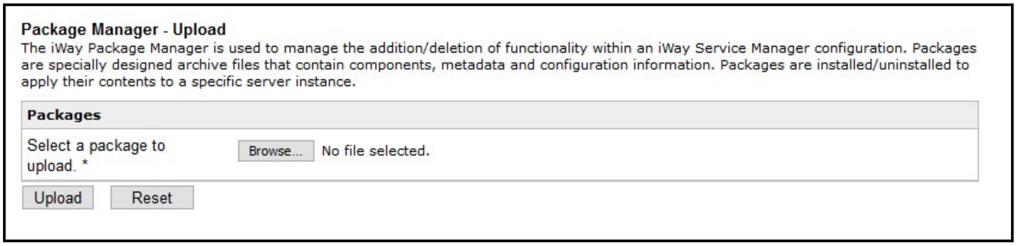
Packages

□ Filter By Name Where Name Equals

□	Filename	Download	Last Modified
□	iwhl7llp-package.zip		Mar 13, 2007 2:28:46 PM

3. Click *Add* to proceed to the Package upload page.

You will be prompted for the package to upload, as shown in the following image. You may either enter the fully qualified file name or click *Browse* to navigate to the file's location.



Package Manager - Upload

The iWay Package Manager is used to manage the addition/deletion of functionality within an iWay Service Manager configuration. Packages are specially designed archive files that contain components, metadata and configuration information. Packages are installed/uninstalled to apply their contents to a specific server instance.

Packages	
Select a package to upload. *	Browse... No file selected.

4. Click *Browse* to see a list of packages.

   Packages must exist on your system ready to be uploaded. Alternatively, you can copy packages into the following directory:

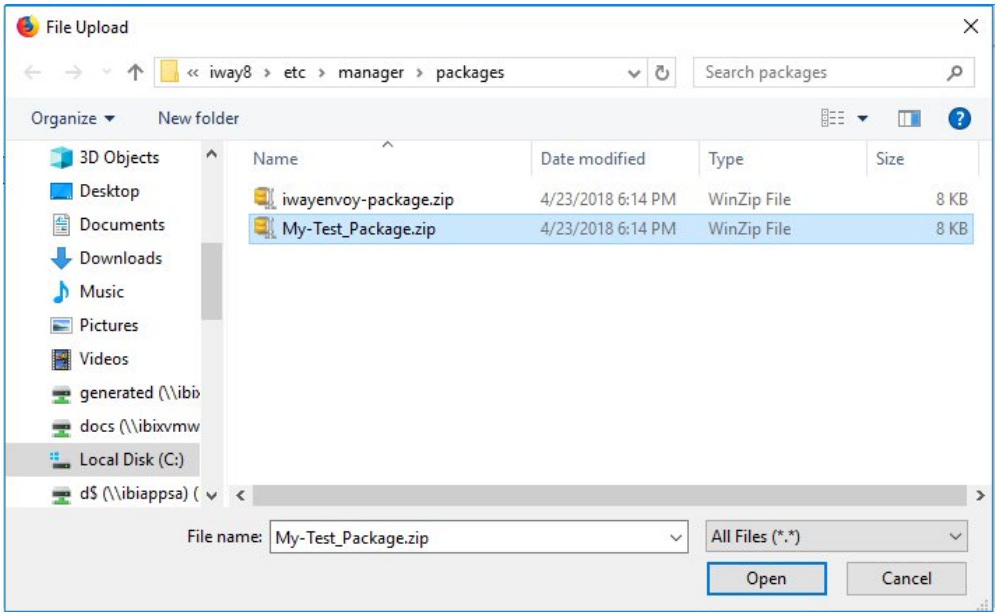
   ```
   <iwayhome>\etc\manager\packages
   ```

   where:

   <iwayhome>

   Is the location on your system where iWay Service Manager is installed.

   In this example, the package file, My-Test-Package.zip, is available for upload, as shown in the following image.



File Upload

« iway8 › etc › manager › packages

Name	Date modified	Type	Size
iwayenvoy-package.zip	4/23/2018 6:14 PM	WinZip File	8 KB
My-Test_Package.zip	4/23/2018 6:14 PM	WinZip File	8 KB

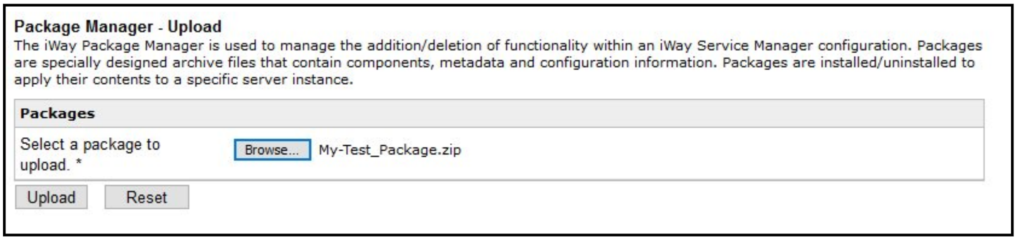
File name: My-Test_Package.zip

All Files (*.*)

Note: The actual appearance of the Choose File window will depend on your browser.

5. Select the package to upload, and click *Open*.

   The chosen file name populates the upload field, as shown in the following image.



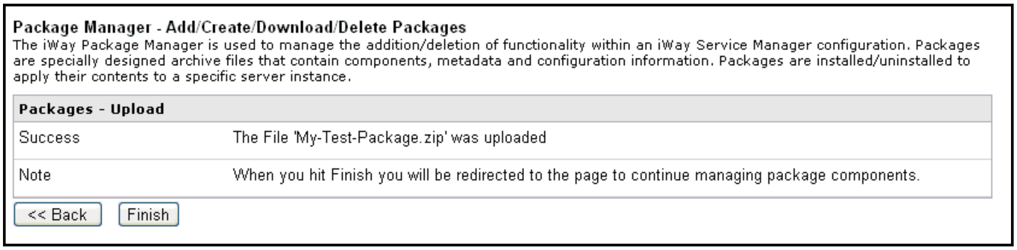
Package Manager - Upload

The iWay Package Manager is used to manage the addition/deletion of functionality within an iWay Service Manager configuration. Packages are specially designed archive files that contain components, metadata and configuration information. Packages are installed/uninstalled to apply their contents to a specific server instance.

Packages	
Select a package to upload. *	Browse... My-Test_Package.zip

6. Click *Upload*.

   The following image shows a Success confirmation page of the uploaded package and a note to click the Finish button to continue managing package components.



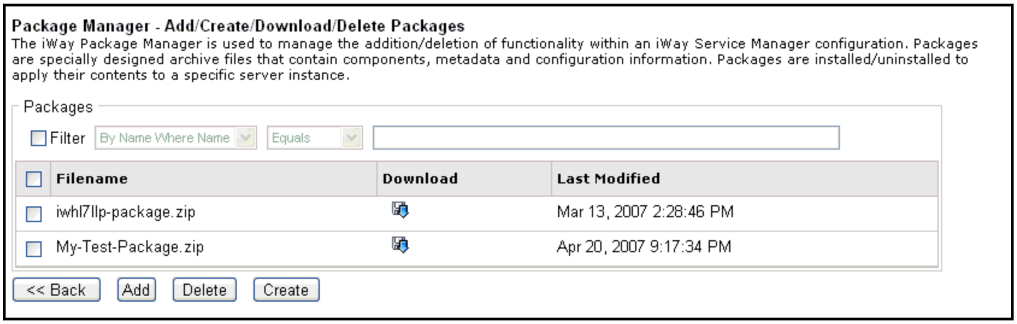
Package Manager - Add/Create/Download/Delete Packages

The iWay Package Manager is used to manage the addition/deletion of functionality within an iWay Service Manager configuration. Packages are specially designed archive files that contain components, metadata and configuration information. Packages are installed/uninstalled to apply their contents to a specific server instance.

Packages - Upload	
Success	The File 'My-Test-Package.zip' was uploaded
Note	When you hit Finish you will be redirected to the page to continue managing package components.

7. Click *Finish* to return to the list of available packages.

   The newly uploaded package name appears on the list of available packages, as shown in the following image.



Package Manager - Add/Create/Download/Delete Packages

The iWay Package Manager is used to manage the addition/deletion of functionality within an iWay Service Manager configuration. Packages are specially designed archive files that contain components, metadata and configuration information. Packages are installed/uninstalled to apply their contents to a specific server instance.

Packages

☐ Filter By Name Where Name Equals

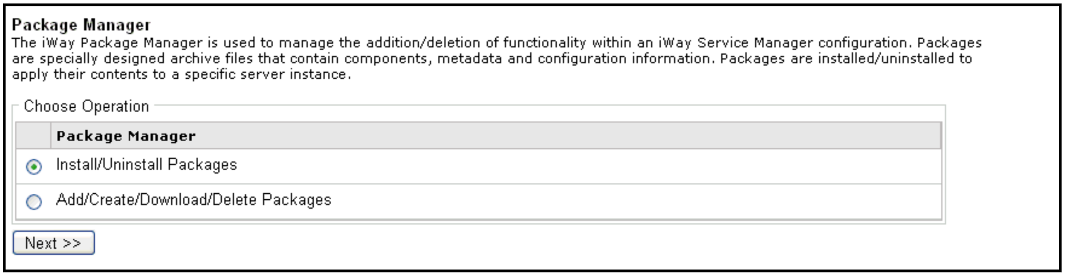
☐	Filename	Download	Last Modified
☐	iwhl7llp-package.zip	⬇	Mar 13, 2007 2:28:46 PM
☐	My-Test-Package.zip	⬇	Apr 20, 2007 9:17:34 PM

Note: At this time your package is not installed, but has been made available to the Package Installation process.

Installing a Package

iWay Package files contain a set of related iWay Service Manager components to enable a particular enterprise integration objective. When you install a package on a managed server, you are in effect installing all of the components contained in that package. The exception is when there is duplication of the components (for example, one or more components are previously installed). It is at your discretion whether the old version is overwritten or preserved. Installing a package affects the runtime instance configuration directly.

The following image shows the Package Manager, which lists two operations you can choose from, Install/Uninstall Packages and Add/Create/Download/Delete Packages.



Package Manager

The iWay Package Manager is used to manage the addition/deletion of functionality within an iWay Service Manager configuration. Packages are specially designed archive files that contain components, metadata and configuration information. Packages are installed/uninstalled to apply their contents to a specific server instance.

Choose Operation

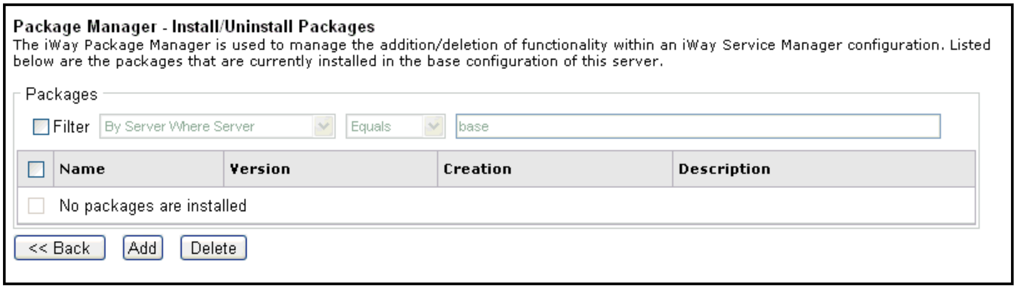
	Package Manager
●	Install/Uninstall Packages
○	Add/Create/Download/Delete Packages

Procedure: How to Install a Package

To install a package:

1. Select the *Install/Uninstall Packages* button, and click *Next*.

   As shown in the following image, Package Manager displays that no packages are installed.



Package Manager - Install/Uninstall Packages

The iWay Package Manager is used to manage the addition/deletion of functionality within an iWay Service Manager configuration. Listed below are the packages that are currently installed in the base configuration of this server.

Packages

□ Filter By Server Where Server | Equals | base

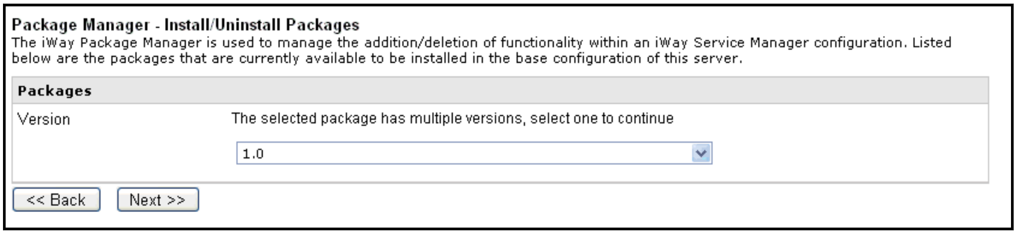
□	Name	Version	Creation	Description
□	No packages are installed			

2. Click *Add* to proceed to the list of available packages.

iWay Service Manager is shipped with the iwhl7llp package, which adds support for the HL7 MLLP protocol. You can choose which packages will be copied to your server during the iWay Service Manager installation. This page shows you the packages that are available for installation on the server including those transferred at installation time and any you have manually copied to [IWAYHOME]/etc/manager/packages in your server file system.

3. Select the package you want to install and click *Next*.

   The version pane for the selected package opens.



Package Manager - Install/Uninstall Packages

The iWay Package Manager is used to manage the addition/deletion of functionality within an iWay Service Manager configuration. Listed below are the packages that are currently available to be installed in the base configuration of this server.

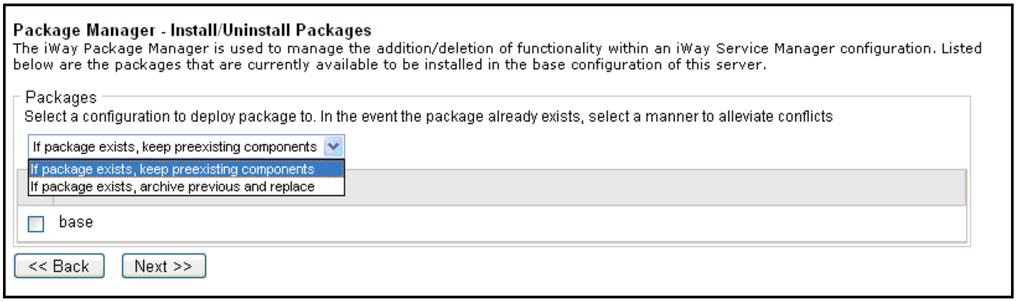
Packages

Version: The selected package has multiple versions, select one to continue

1.0

4. Choose one of the listed versions and click *Next*.

   The Configuration page allows you to select the target configuration for the package and how to apply the package if there are pre-existing components. If the package exists in the selected configuration, you may choose to preserve pre-existing components or to overwrite and save any pre-existing components.



Package Manager - Install/Uninstall Packages

The iWay Package Manager is used to manage the addition/deletion of functionality within an iWay Service Manager configuration. Listed below are the packages that are currently available to be installed in the base configuration of this server.

Packages

Select a configuration to deploy package to. In the event the package already exists, select a manner to alleviate conflicts

If package exists, keep preexisting components

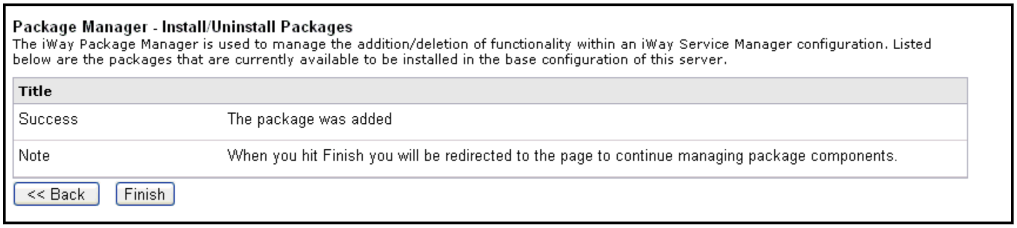
If package exists, keep preexisting components
If package exists, archive previous and replace

□ base

5. Select the server configuration you wish to install the package to, and the method to handle pre-existing components.

6. Click the box for the base configuration, and click *Next*.

The Package installation status page displays that the package was successfully added, as shown in the following image.



Package Manager - Install/Uninstall Packages

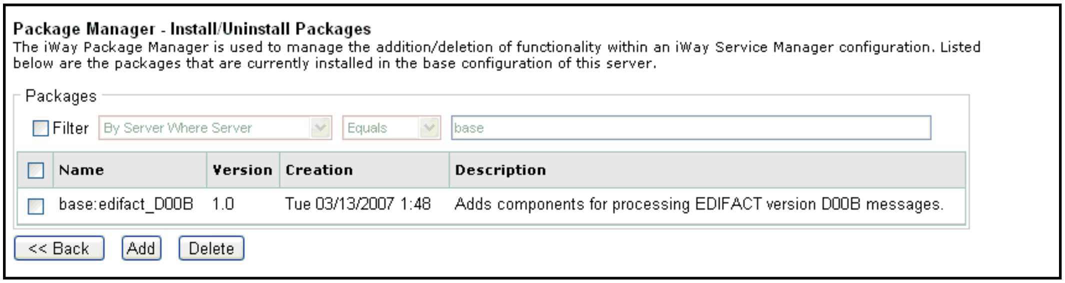
The iWay Package Manager is used to manage the addition/deletion of functionality within an iWay Service Manager configuration. Listed below are the packages that are currently available to be installed in the base configuration of this server.

Title	
Success	The package was added
Note	When you hit Finish you will be redirected to the page to continue managing package components.

7. Click *Finish*.

The server must be restarted for the components, providers, and SREGS to become available.

As shown in the following image, the list of packages reflects the result of this process, and the newly added EDIFACT package is listed as installed in the base configuration.



Package Manager - Install/Uninstall Packages

The iWay Package Manager is used to manage the addition/deletion of functionality within an iWay Service Manager configuration. Listed below are the packages that are currently installed in the base configuration of this server.

Packages

□ Filter By Server Where Server | Equals | base

□	Name	Version	Creation	Description
□	base:edifact_D00B	1.0	Tue 03/13/2007 1:48	Adds components for processing EDIFACT version D00B messages.

Uninstalling a Package

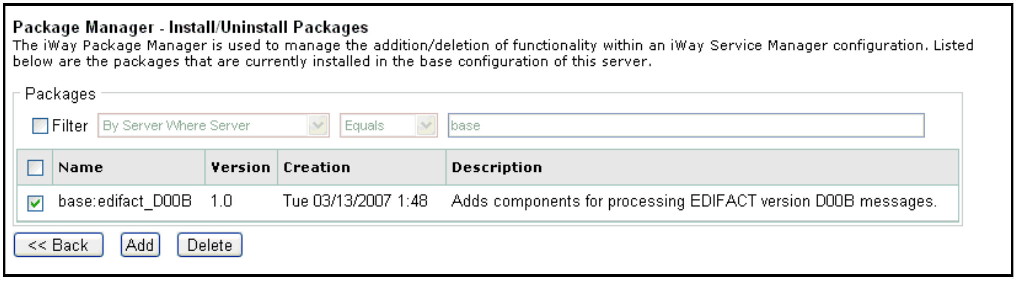
There are two scenarios that are described on-screen as "deleting a package." The first is the removal of a package's installed components from a configuration. This will affect the run-time behavior of the server, since functionality is removed.

In the second case, a package file is deleted from the host server’s file system. This does not directly affect run-time behavior, but will make the package unavailable for future installation. For more information on deleting a package, see *Deleting a Package* on page 364.

Procedure: How to Uninstall a Package

1. Select the package to be deleted.

The following image shows a sample Package Manager - Install/Uninstall Packages page with the base:edifact_DOOB check box selected.



Package Manager - Install/Uninstall Packages

The iWay Package Manager is used to manage the addition/deletion of functionality within an iWay Service Manager configuration. Listed below are the packages that are currently installed in the base configuration of this server.

Packages

□ Filter | By Server Where Server | Equals | base

□	Name	Version	Creation	Description
☑	base:edifact_D00B	1.0	Tue 03/13/2007 1:48	Adds components for processing EDIFACT version D00B messages.

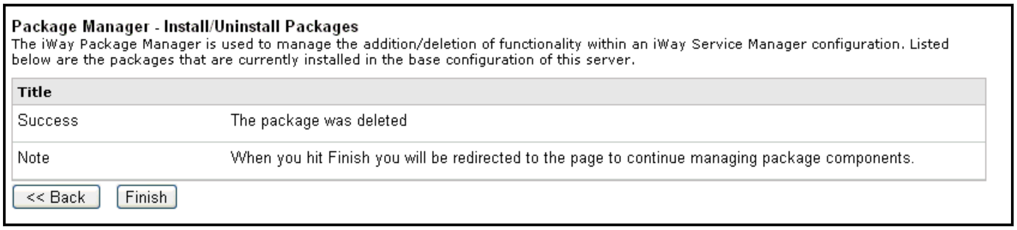
2. Click *Delete*.

   A confirmation dialog box opens, prompting you to confirm the deletion of the package.

3. Click *OK* if you wish to delete the package.

   Note: Deleting a package is permanent. There is no *undo* (short of re-installing the package) if you mistakenly delete a package.

   The following image shows a Success confirmation page of the deleted package and a note to click the Finish button to continue managing package components.



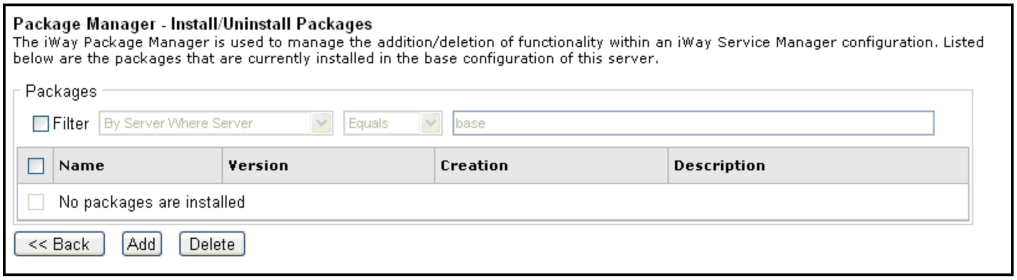
Package Manager - Install/Uninstall Packages

The iWay Package Manager is used to manage the addition/deletion of functionality within an iWay Service Manager configuration. Listed below are the packages that are currently installed in the base configuration of this server.

Title	
Success	The package was deleted
Note	When you hit Finish you will be redirected to the page to continue managing package components.

4. Click *Finish*.

   As shown in the following image, there is no package installed, further confirming that the base:edifact component was deleted.



Package Manager - Install/Uninstall Packages

The iWay Package Manager is used to manage the addition/deletion of functionality within an iWay Service Manager configuration. Listed below are the packages that are currently installed in the base configuration of this server.

Packages

□ Filter | By Server Where Server | Equals | base

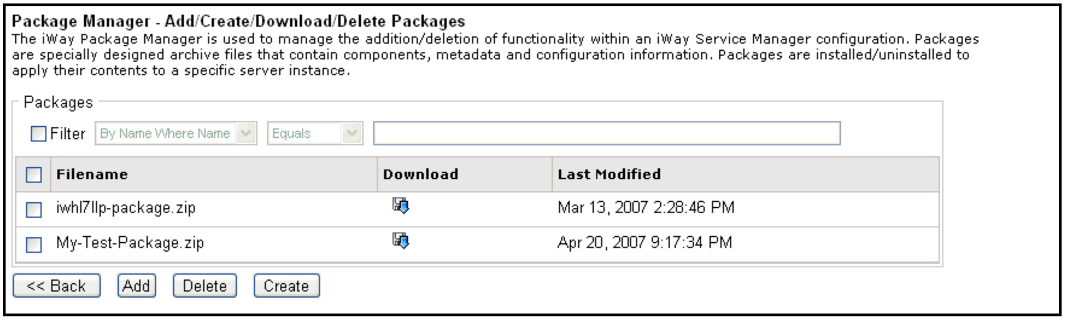
□	Name	Version	Creation	Description
□	No packages are installed			

Creating a Package

You can create a package to enable the exporting of components to another managed server/ repository on the same host or a remote host. For example, you can *clone* part of a configuration, export it, and then import it to a different server for execution.

Note: Creating archives instead of packages is recommended when migrating between different environments (for example, development, test, and production). For more information on how to create an archive, see *Using the Archive Manager* on page 365.

The following image shows the Package Manager - Add/Create/Download/Delete Packages pane.



Package Manager - Add/Create/Download/Delete Packages

The iWay Package Manager is used to manage the addition/deletion of functionality within an iWay Service Manager configuration. Packages are specially designed archive files that contain components, metadata and configuration information. Packages are installed/uninstalled to apply their contents to a specific server instance.

Packages

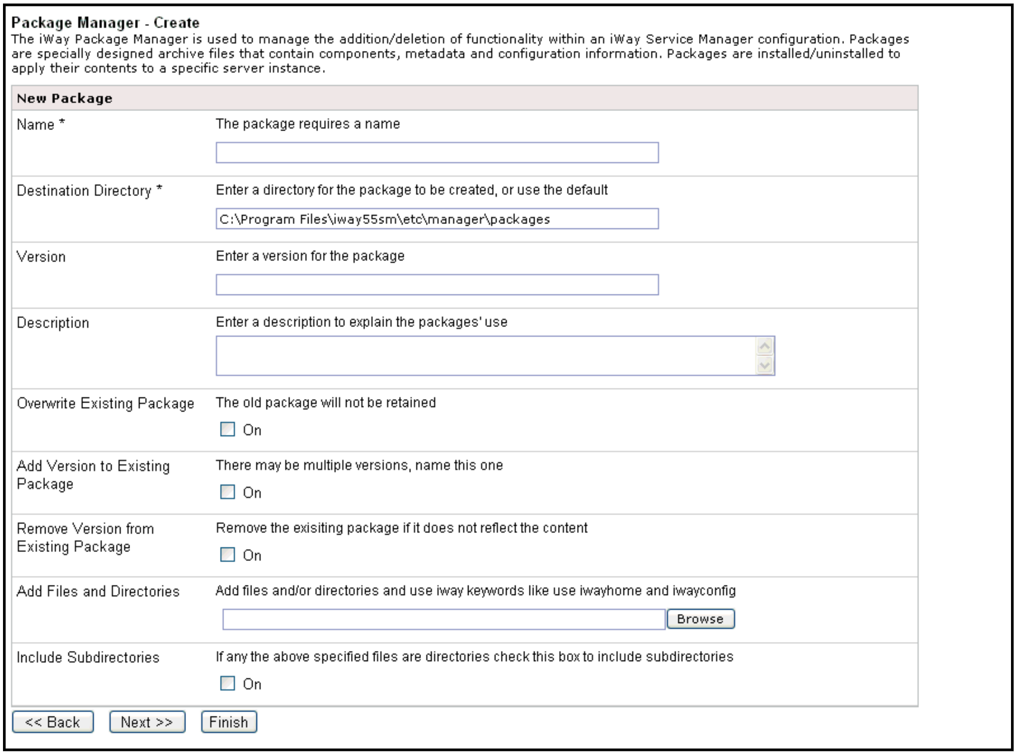
□ Filter By Name Where Name Equals

□	Filename	Download	Last Modified
□	iwhl7llp-package.zip		Mar 13, 2007 2:28:46 PM
□	My-Test-Package.zip		Apr 20, 2007 9:17:34 PM

Procedure: How to Create a Package

1. Click *Create*.

   The Package Manager - Create window opens. The following image shows the New Package page, where you provide basic information about the new package.



Package Manager - **Create**

The iWay Package Manager is used to manage the addition/deletion of functionality within an iWay Service Manager configuration. Packages are specially designed archive files that contain components, metadata and configuration information. Packages are installed/uninstalled to apply their contents to a specific server instance.

New Package	
Name *	The package requires a name
Destination Directory *	Enter a directory for the package to be created, or use the default C:\Program Files\iway55sm\etc\manager\packages
Version	Enter a version for the package
Description	Enter a description to explain the packages' use
Overwrite Existing Package	The old package will not be retained □ On
Add Version to Existing Package	There may be multiple versions, name this one □ On
Remove Version from Existing Package	Remove the exisiting package if it does not reflect the content □ On
Add Files and Directories	Add files and/or directories and use iway keywords like use iwayhome and iwayconfig Browse
Include Subdirectories	If any the above specified files are directories check this box to include subdirectories □ On

The following table lists and describes the properties for this page.

Property	Description
Name	Name of the package to create.
Destination Directory	Where the package file will be written upon creation.
Version	Version of the package.
Description	Brief description of the package.

Property	Description
Overwrite Existing Package	If checked, the new package will overwrite any preexisting one with the same name.
Add Version to Existing Package	Add this version to the existing package.
Remove Version from Existing Package	Remove the existing package.
Add Files and Directories	Add component files and/or directories to the package file (these directories are relative to and accessible by the iWay server, not your browser).
Include Subdirectories	If the above option is selected, include subdirectories when adding files and/or directories.

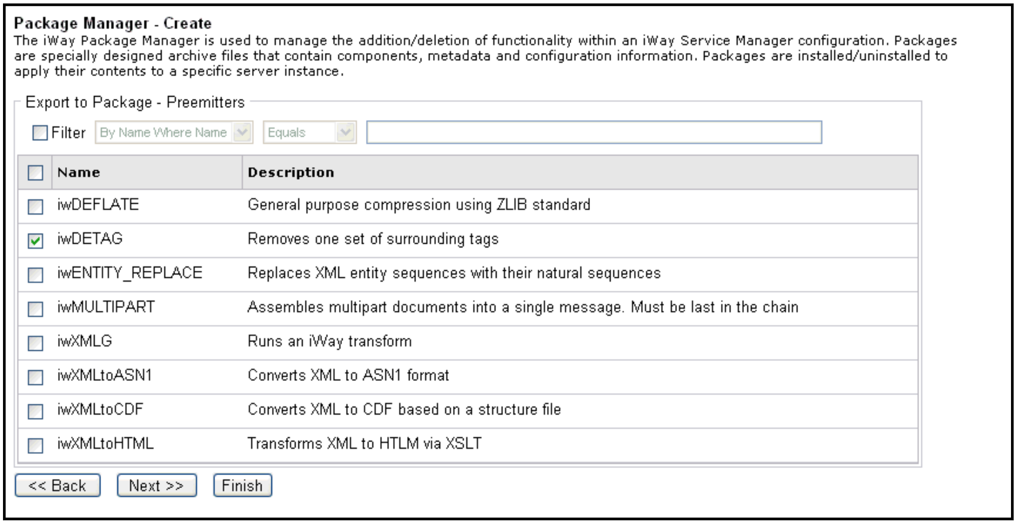
2. Click *Next* to continue to a series of component selection pages. Component selection pages are presented in the following order:

   - ❏ Adapters
   - ❏ Agents
   - ❏ Preemitters
   - ❏ Emitters
   - ❏ Preparsers
   - ❏ Encryptors
   - ❏ Ebix
   - ❏ Reviewers
   - ❏ Exits
   - ❏ Transforms
   - ❏ Validation
   - ❏ Schemas
   - ❏ Stored Procedures

- ❑ PFlows (Process Flows)
- ❑ Listeners
- ❑ Global Documents
- ❑ Special Registers
- ❑ JLINK Data Providers
- ❑ JDBC Data Providers
- ❑ LDAP Providers Keystore Providers
- ❑ SSL Context Providers
- ❑ Directory Certstore Providers
- ❑ LDAP Certstore Providers
- ❑ OCSP Responser Providers
- ❑ Signature Policy Providers
- ❑ Namespace Mapping Providers
- ❑ HTTP Client Providers
- ❑ Authentication Realm Providers
- ❑ Scheduler Providers
- ❑ SNMP Providers
- ❑ DQ Providers
- ❑ DQ Runtime Providers
- ❑ Activity Facilities
- ❑ Correlation Facilities
- ❑ Trading Partner Access Handlers

3. Select the check box for the component you want to export.

The following image shows the Export to Package - Preemitters page, which lists the components. The iwDETAG check box is selected.



Package Manager - Create

The iWay Package Manager is used to manage the addition/deletion of functionality within an iWay Service Manager configuration. Packages are specially designed archive files that contain components, metadata and configuration information. Packages are installed/uninstalled to apply their contents to a specific server instance.

Export to Package - Preemitters

□ Filter By Name Where Name | Equals

□	Name	Description
□	iwDEFLATE	General purpose compression using ZLIB standard
☑	iwDETAG	Removes one set of surrounding tags
□	iwENTITY_REPLACE	Replaces XML entity sequences with their natural sequences
□	iwMULTIPART	Assembles multipart documents into a single message. Must be last in the chain
□	iwXMLG	Runs an iWay transform
□	iwXMLtoASN1	Converts XML to ASN1 format
□	iwXMLtoCDF	Converts XML to CDF based on a structure file
□	iwXMLtoHTML	Transforms XML to HTLM via XSLT

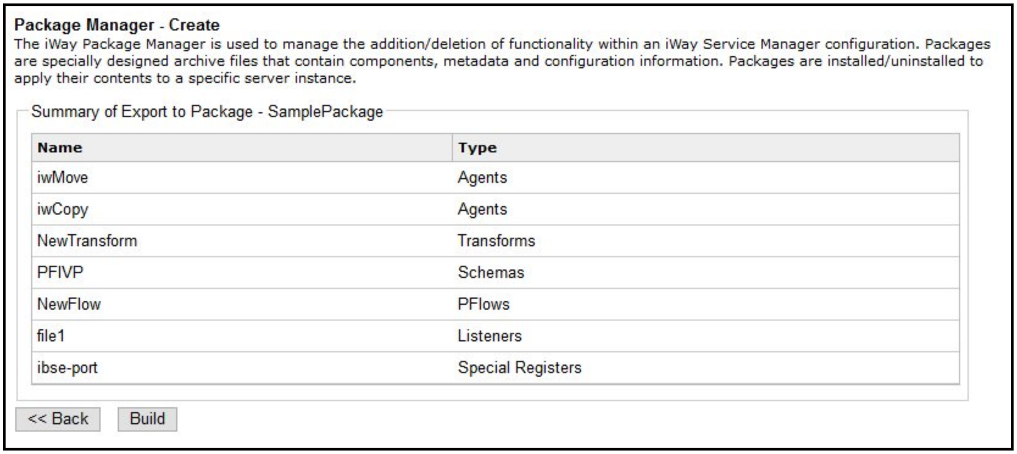
It is the user's responsibility to export all related components (including dependencies), so that the finished package is complete and consistent.

4. Click *Next* to proceed to the next component selection page. A list showing the order of component selection pages is available in Step 2.

   Note: From any of the component selection pages, you may click *Finish* to skip to the Create Package summary page.

5. Click *Finish*.

The following image shows a summary page of the package information and selected components for export.



Package Manager - Create

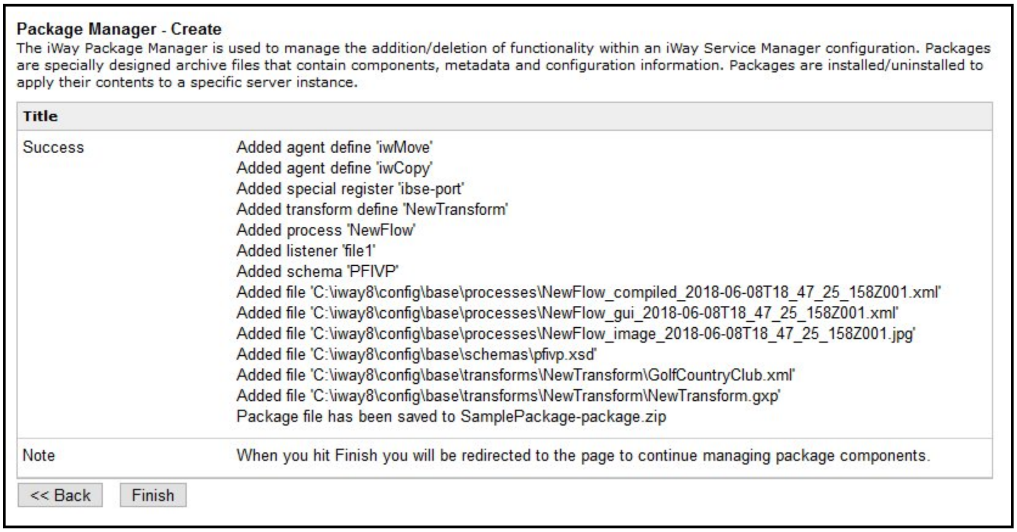
The iWay Package Manager is used to manage the addition/deletion of functionality within an iWay Service Manager configuration. Packages are specially designed archive files that contain components, metadata and configuration information. Packages are installed/uninstalled to apply their contents to a specific server instance.

Summary of Export to Package - SamplePackage

Name	Type
iwMove	Agents
iwCopy	Agents
NewTransform	Transforms
PFIVP	Schemas
NewFlow	PFlows
file1	Listeners
ibse-port	Special Registers

6. Click *Finish*.

   The following image shows a status page indicating that the agent and preemitter components were successfully added to the Detag_export-package.zip file.



Package Manager - Create

The iWay Package Manager is used to manage the addition/deletion of functionality within an iWay Service Manager configuration. Packages are specially designed archive files that contain components, metadata and configuration information. Packages are installed/uninstalled to apply their contents to a specific server instance.

Title	
Success	Added agent define 'iwMove' Added agent define 'iwCopy' Added special register 'ibse-port' Added transform define 'NewTransform' Added process 'NewFlow' Added listener 'file1' Added schema 'PFIVP' Added file 'C:\iway8\config\base\processes\NewFlow_compiled_2018-06-08T18_47_25_158Z001.xml' Added file 'C:\iway8\config\base\processes\NewFlow_gui_2018-06-08T18_47_25_158Z001.xml' Added file 'C:\iway8\config\base\processes\NewFlow_image_2018-06-08T18_47_25_158Z001.jpg' Added file 'C:\iway8\config\base\schemas\pfivp.xsd' Added file 'C:\iway8\config\base\transforms\NewTransform\GolfCountryClub.xml' Added file 'C:\iway8\config\base\transforms\NewTransform\NewTransform.gxp' Package file has been saved to SamplePackage-package.zip
Note	When you hit Finish you will be redirected to the page to continue managing package components.

7. Click *Finish* to return to the Package listing page.

   The new package, Detag_export-package.zip, is added to the list of pre-existing package files.

Downloading a Package

You may wish to copy a package from your server to your workstation for backup, in preparation for an installation to another server, or to email it.

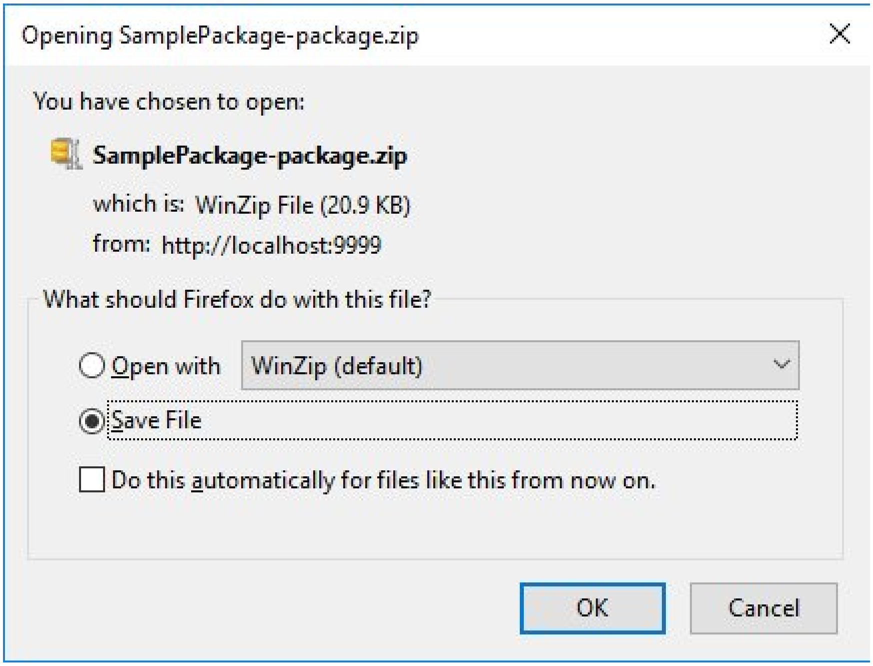
Procedure: How to Download a Package

1. Click the *Download* icon which corresponds to the package you wish to download.

   The File Download dialog box is displayed, asking whether to open or save the selected file.

2. Click *Save* to download the file.

   The following image shows the Save As dialog box with the selected file in the File name field.



Opening SamplePackage-package.zip

You have chosen to open:

SamplePackage-package.zip

which is: WinZip File (20.9 KB)

from: http://localhost:9999

What should Firefox do with this file?

○ Open with WinZip (default)

● Save File

□ Do this automatically for files like this from now on.

   Note: The actual appearance of the dialog box will depend on your browser.

3. Specify the location and click *Save*.

   The browser will copy the file to the specified location.

Deleting a Package

You may wish to delete a package to clean up the server or shorten the list of package files being displayed.

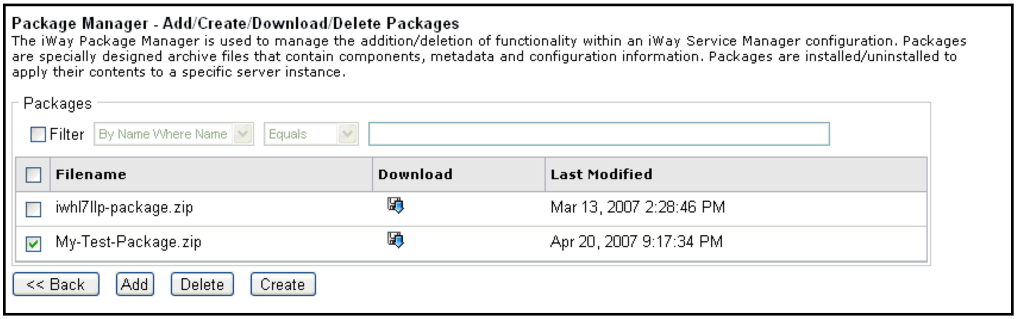
Caution: Deleting a package actually deletes it from the host server's file system. This does not directly affect run-time behavior, but will make the package unavailable for future installation.

You can also uninstall a package from the configuration. Uninstalling a package removes the package from the configuration, but leaves it on the file system on the host server. For more information, see *Uninstalling a Package* on page 355.

Procedure: How to Delete a Package

1. Select the package to be deleted.

   The following image shows the Package Manager - Add/Create/Download/Delete Packages page which lists the available packages for deletion with the My-Test-Package.zip file selected.



Package Manager - Add/Create/Download/Delete Packages

The iWay Package Manager is used to manage the addition/deletion of functionality within an iWay Service Manager configuration. Packages are specially designed archive files that contain components, metadata and configuration information. Packages are installed/uninstalled to apply their contents to a specific server instance.

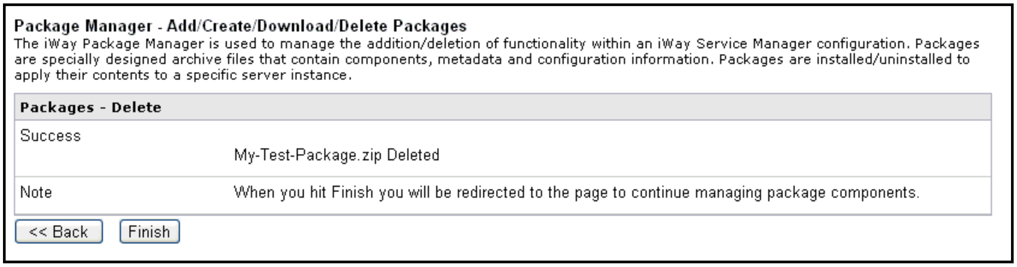
Packages

☐ Filter | By Name Where Name | Equals |

☐	Filename	Download	Last Modified
☐	iwhl7llp-package.zip		Mar 13, 2007 2:28:46 PM
☑	My-Test-Package.zip		Apr 20, 2007 9:17:34 PM

2. Click *Delete*.

   The status page is displayed, as shown in the following image, confirming that the .zip file has been deleted.



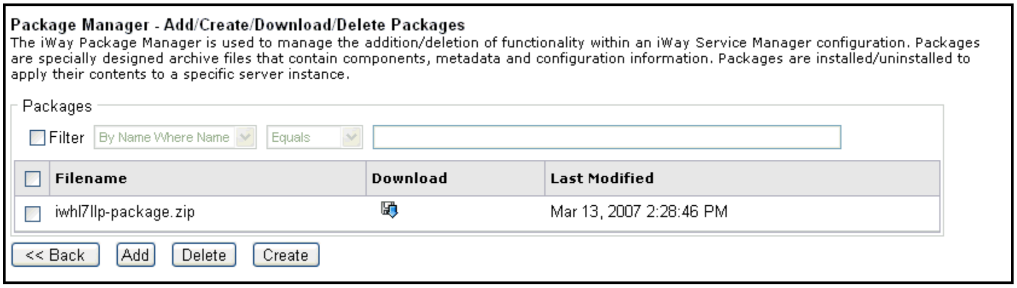
Package Manager - Add/Create/Download/Delete Packages

The iWay Package Manager is used to manage the addition/deletion of functionality within an iWay Service Manager configuration. Packages are specially designed archive files that contain components, metadata and configuration information. Packages are installed/uninstalled to apply their contents to a specific server instance.

Packages - Delete	
Success	My-Test-Package.zip Deleted
Note	When you hit Finish you will be redirected to the page to continue managing package components.

3. Click *Finish* to continue to the list of available package files.

As shown in the following image, the My-Test-Package.zip is not listed as an available archive file, confirming its deletion.



Package Manager - Add/Create/Download/Delete Packages

The iWay Package Manager is used to manage the addition/deletion of functionality within an iWay Service Manager configuration. Packages are specially designed archive files that contain components, metadata and configuration information. Packages are installed/uninstalled to apply their contents to a specific server instance.

□	Filename	Download	Last Modified
□	iwhl7llp-package.zip		Mar 13, 2007 2:28:46 PM

Using the Archive Manager

The iWay Archive Manager is used to import components into the registry from archives or available iSM configurations. It is helpful when you are required to move components (for example, channels or an entire registry) between servers. Archives are specially designed files that contain components, metadata, and configuration information from runtime.

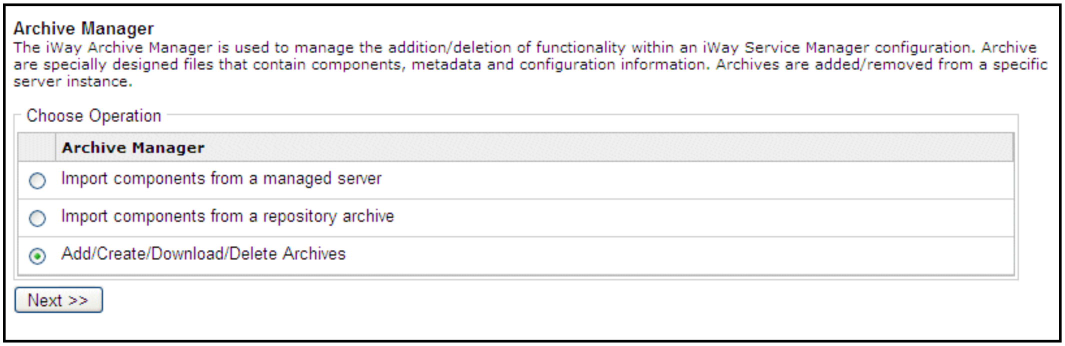
Note: For additional archiving recommendations, see the *Release Notes* or the *iWay Service Manager Migration Guide*.

To access the Archive Manager, click *Archive Manager* in the left console pane of the Tools menu, as shown in the following image.

Adding, Creating, Downloading, and Deleting Archives

To import components from an archive file, the file must be in a predefined location within the server machine's file system. Add will take an archive from any location accessible by your browser and copy it to the appropriate directory on the server machine.

The following image shows the list of operations you can select from the Archive Manager, with the Add/Create/Download/Delete Archives button selected.



Archive Manager

The iWay Archive Manager is used to manage the addition/deletion of functionality within an iWay Service Manager configuration. Archive are specially designed files that contain components, metadata and configuration information. Archives are added/removed from a specific server instance.

Choose Operation

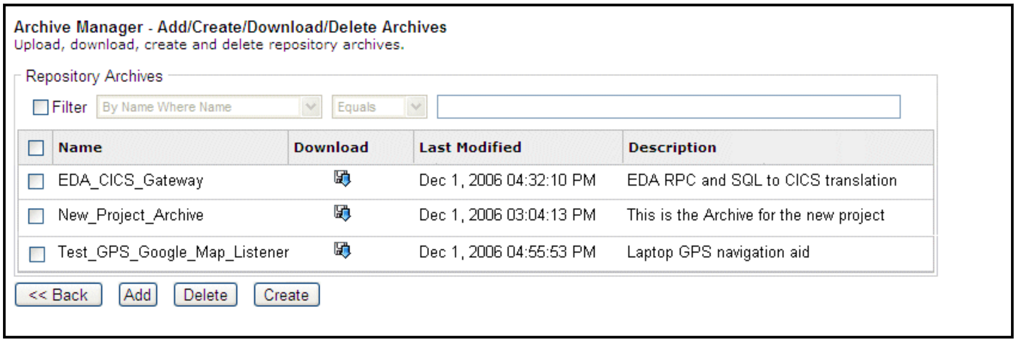
	Archive Manager
○	Import components from a managed server
○	Import components from a repository archive
●	Add/Create/Download/Delete Archives

Procedure: How to Add an Archive

From the Archive Manager:

1. Select the *Add/Create/Download/Delete Archives* button. Click *Next* to continue.

   The list of uploaded archive files will be displayed. As shown in the following image, there are three archive files to choose from, EDA_CICS_Gateway, New_Project_Archive, and Test_GPS_Google_Map_Listener.



Archive Manager - Add/Create/Download/Delete Archives

Upload, download, create and delete repository archives.

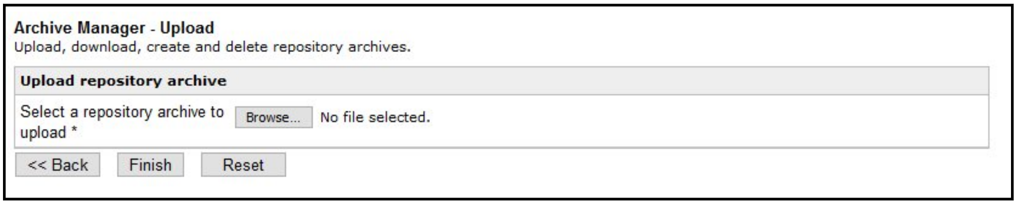
Repository Archives

□ Filter By Name Where Name Equals

□	Name	Download	Last Modified	Description
□	EDA_CICS_Gateway		Dec 1, 2006 04:32:10 PM	EDA RPC and SQL to CICS translation
□	New_Project_Archive		Dec 1, 2006 03:04:13 PM	This is the Archive for the new project
□	Test_GPS_Google_Map_Listener		Dec 1, 2006 04:55:53 PM	Laptop GPS navigation aid

2. Click *Add* to proceed to the Archive upload page.

   As shown in the following image, you will be prompted for the archive file to upload on the Archive Manager - Upload page. You may either enter the fully qualified file name or click *Browse* to navigate to the file's location.



Archive Manager - Upload

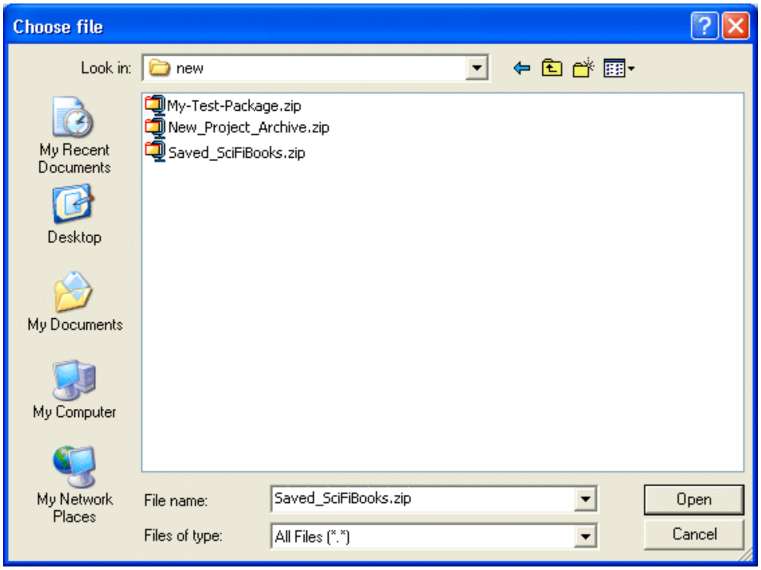
Upload, download, create and delete repository archives.

Upload repository archive

Select a repository archive to upload * No file selected.

3. Click *Browse* to find the location of the archive.

   In this example the archive file Saved_SciFiBooks.zip is chosen for upload from a local directory on a system, as shown in the following image.



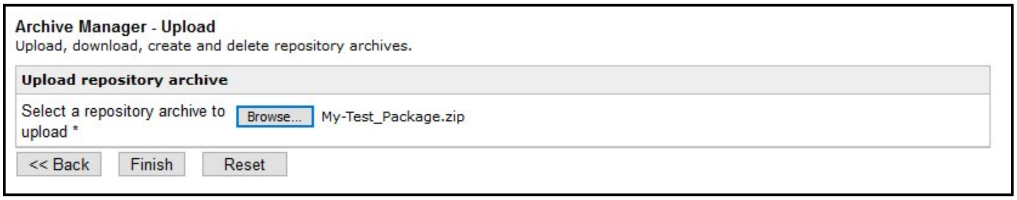
File name: Saved_SciFiBooks.zip

Files of type: All Files (*.*)

Note: The actual appearance of the Choose File window will depend on your browser.

4. Select the archive to upload, and click *Open*.

   The chosen file name will populate in the upload field, as shown in the following image.



Archive Manager - Upload

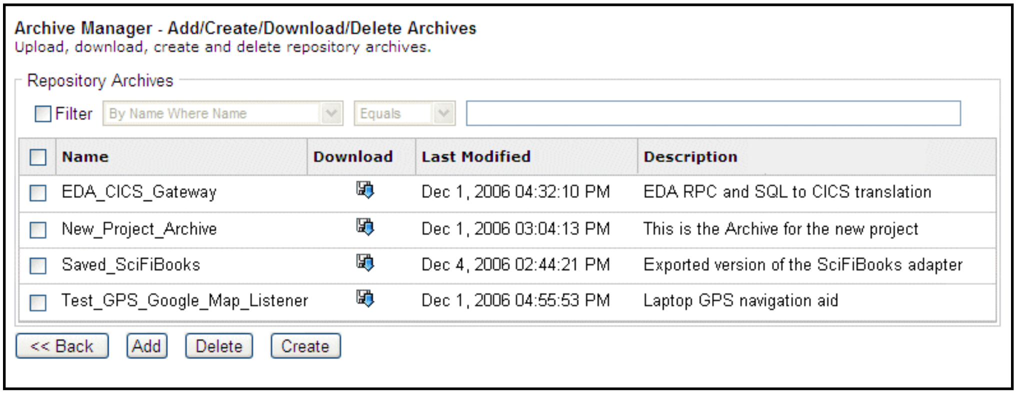
Upload, download, create and delete repository archives.

Upload repository archive

Select a repository archive to upload * My-Test_Package.zip

5. Click *Next* to upload the archive.

   The newly uploaded archive name is added to the list of available archives, as shown in the following image.



Archive Manager - Add/Create/Download/Delete Archives

Upload, download, create and delete repository archives.

Repository Archives

□ Filter By Name Where Name Equals

□	Name	Download	Last Modified	Description
□	EDA_CICS_Gateway		Dec 1, 2006 04:32:10 PM	EDA RPC and SQL to CICS translation
□	New_Project_Archive		Dec 1, 2006 03:04:13 PM	This is the Archive for the new project
□	Saved_SciFiBooks		Dec 4, 2006 02:44:21 PM	Exported version of the SciFiBooks adapter
□	Test_GPS_Google_Map_Listener		Dec 1, 2006 04:55:53 PM	Laptop GPS navigation aid

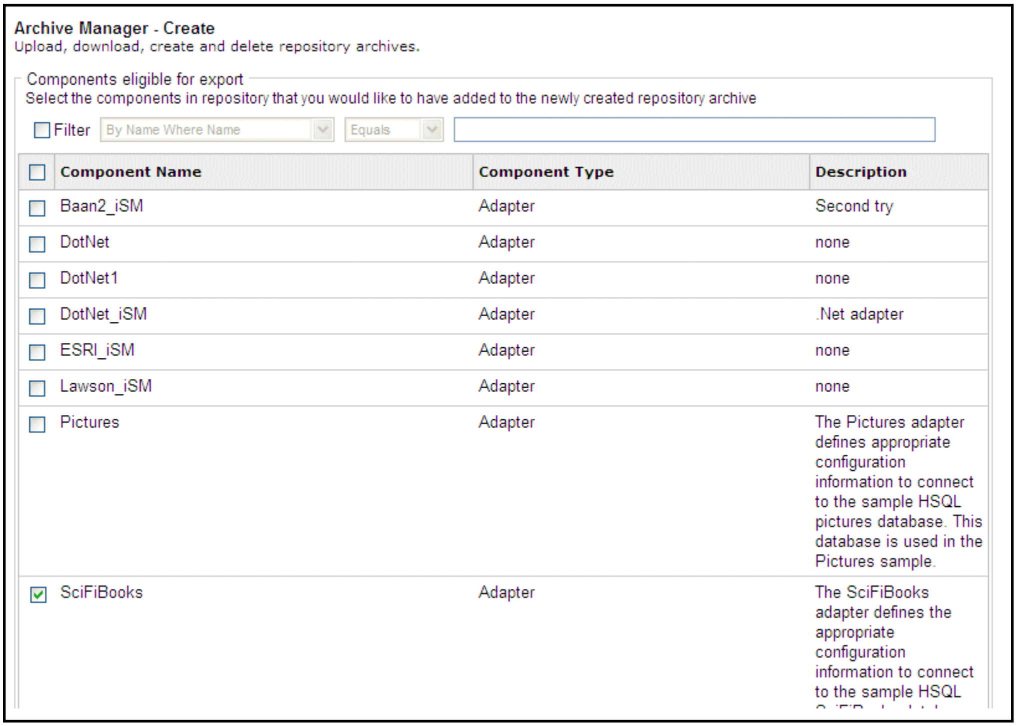
Note: At this time, the components in the archive have not been imported, but just made available to Archive Manager's import component process. You can import the archive at this time.

Procedure: How to Create an Archive

From the list of archive files:

1. Click *Create* to proceed to the list of Registry components which are available for export to an archive file.

This list may be filtered by criteria to minimize its length and help locate the component you are interested in. In this example, the list is not filtered and extends below the browser window, as shown in the following image.



Archive Manager - Create
Upload, download, create and delete repository archives.

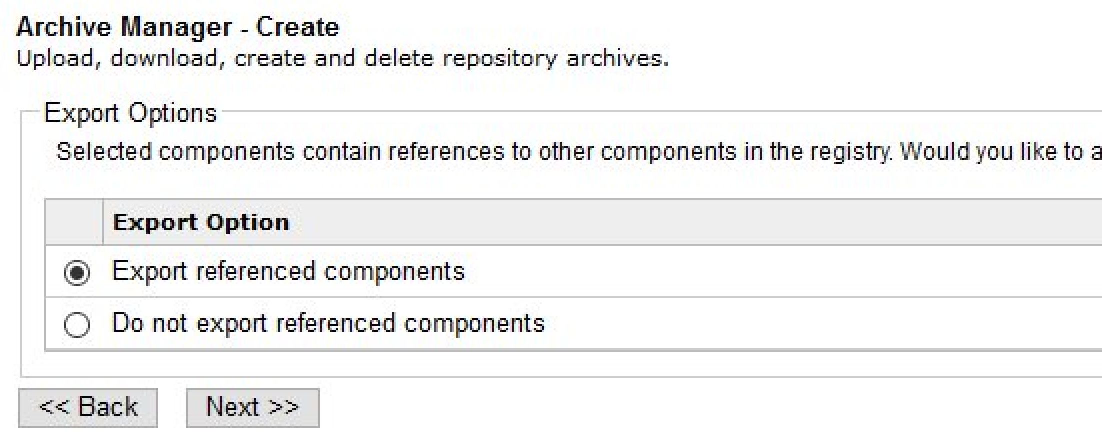
Components eligible for export
Select the components in repository that you would like to have added to the newly created repository archive

☐ Filter By Name Where Name Equals

☐	Component Name	Component Type	Description
☐	Baan2_iSM	Adapter	Second try
☐	DotNet	Adapter	none
☐	DotNet1	Adapter	none
☐	DotNet_iSM	Adapter	.Net adapter
☐	ESRI_iSM	Adapter	none
☐	Lawson_iSM	Adapter	none
☐	Pictures	Adapter	The Pictures adapter defines appropriate configuration information to connect to the sample HSQL pictures database. This database is used in the Pictures sample.
☑	SciFiBooks	Adapter	The SciFiBooks adapter defines the appropriate configuration information to connect to the sample HSQL

2. Select the components you wish to export to the archive file, and click *Next* to continue. In this case, the SciFiBooks adapter is selected for export.

3. Specify whether to export referenced components.



Archive Manager - Create

Upload, download, create and delete repository archives.

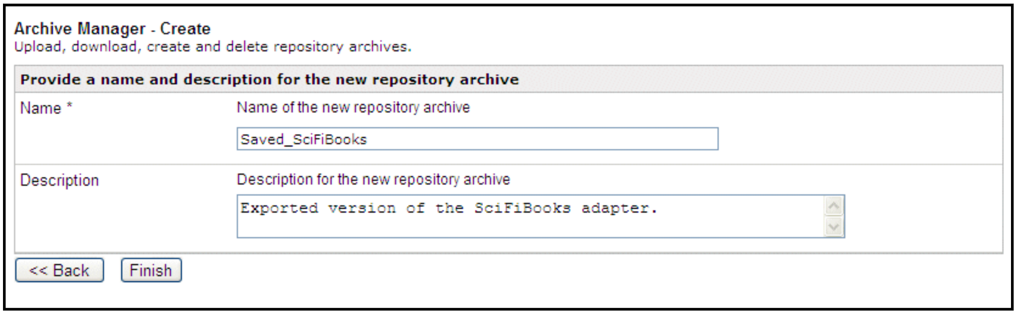
Export Options

Selected components contain references to other components in the registry. Would you like to a

	Export Option
●	Export referenced components
○	Do not export referenced components

4. Enter the name for the archive file and an optional description.

   Archive file names should consist of only letters, numbers, and underscores, and begin with a letter. The following image shows that the SciFiBooks component is named Saved_SciFiBooks and its description is "Exported version of the SciFiBooks adapter."



Archive Manager - Create

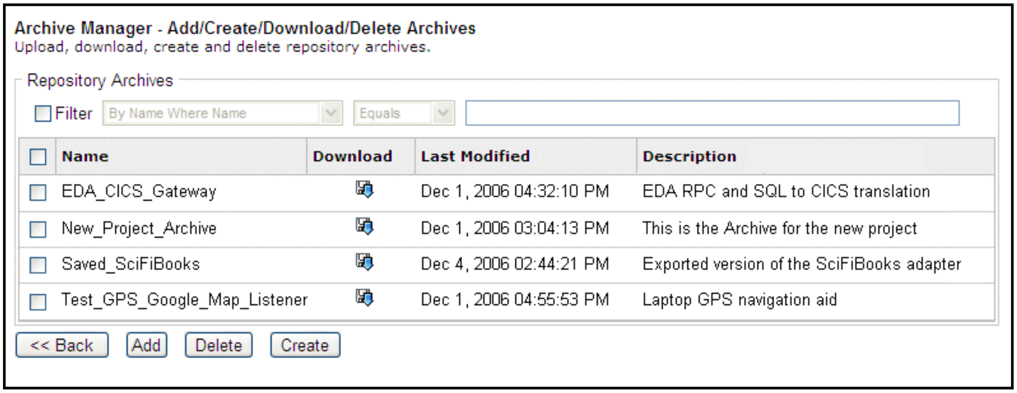
Upload, download, create and delete repository archives.

Provide a name and description for the new repository archive

Name *	Name of the new repository archive Saved_SciFiBooks
Description	Description for the new repository archive Exported version of the SciFiBooks adapter.

5. Click *Finish*.

The Repository Archives listing page displays the newly created archive, as shown in the following image.



Archive Manager - Add/Create/Download/Delete Archives
Upload, download, create and delete repository archives.

Repository Archives

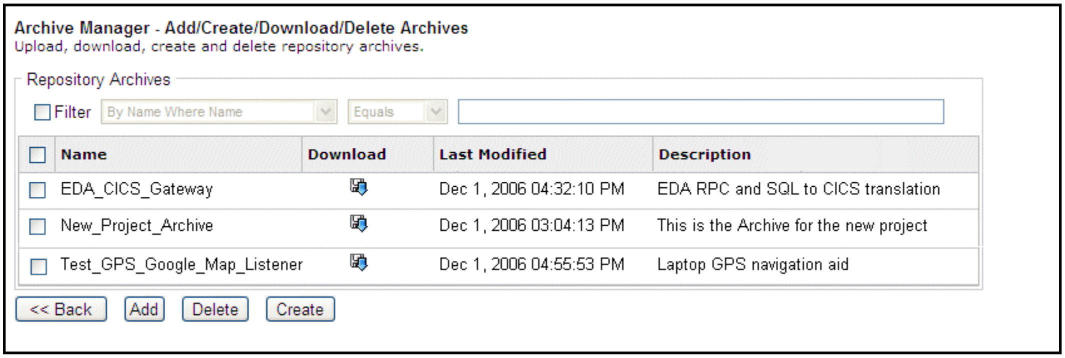
□	Name	Download	Last Modified	Description
□	EDA_CICS_Gateway	⬇	Dec 1, 2006 04:32:10 PM	EDA RPC and SQL to CICS translation
□	New_Project_Archive	⬇	Dec 1, 2006 03:04:13 PM	This is the Archive for the new project
□	Saved_SciFiBooks	⬇	Dec 4, 2006 02:44:21 PM	Exported version of the SciFiBooks adapter
□	Test_GPS_Google_Map_Listener	⬇	Dec 1, 2006 04:55:53 PM	Laptop GPS navigation aid

Procedure: How to Download an Archive

You may wish to copy an archive file from your server to your workstation for backup, in preparation for an installation to another server, or to email it.

Another reason to download an archive is to move it from one system to another.

In the example, the following image shows three packages available for download, EDA_CICS_Gateway, New_Project_Archive, and Test_GPS_Google_Map_Listener.



Archive Manager - Add/Create/Download/Delete Archives
Upload, download, create and delete repository archives.

Repository Archives

□	Name	Download	Last Modified	Description
□	EDA_CICS_Gateway	⬇	Dec 1, 2006 04:32:10 PM	EDA RPC and SQL to CICS translation
□	New_Project_Archive	⬇	Dec 1, 2006 03:04:13 PM	This is the Archive for the new project
□	Test_GPS_Google_Map_Listener	⬇	Dec 1, 2006 04:55:53 PM	Laptop GPS navigation aid

1. Click the *Download* icon next to the package you wish to download (for example, New_Project_Archive).

   The File Download dialog box is displayed, providing the option to open or save the New_Project_Archive.zip package.

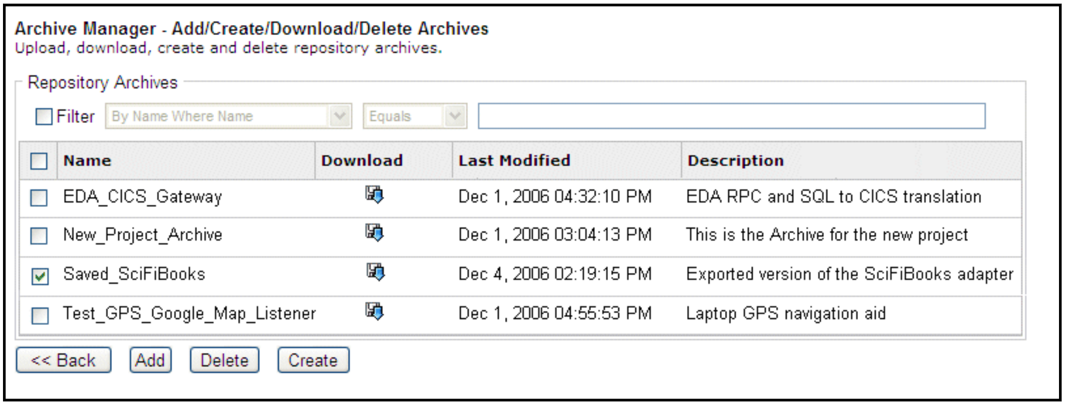
2. To download the file, specify the location and click *Save*.

Note: The dialog box and behavior for saving the file will depend on your browser.

Procedure: How to Delete an Archive

You can use GUI to delete an archive to clean up the server or shorten the list of archive files being displayed.

In the following image, the Archive Manager - Repository Archives page shows four available packages, EDA_CICS_Gateway, New_Project_Archive, Saved_SciFiBooks, and Test_GPS_Google_Map_Listener. The Saved_SciFiBooks archive is selected for deletion.



Archive Manager - Add/Create/Download/Delete Archives

Upload, download, create and delete repository archives.

Repository Archives

□ Filter By Name Where Name Equals

□	Name	Download	Last Modified	Description
□	EDA_CICS_Gateway	⬇	Dec 1, 2006 04:32:10 PM	EDA RPC and SQL to CICS translation
□	New_Project_Archive	⬇	Dec 1, 2006 03:04:13 PM	This is the Archive for the new project
☑	Saved_SciFiBooks	⬇	Dec 4, 2006 02:19:15 PM	Exported version of the SciFiBooks adapter
□	Test_GPS_Google_Map_Listener	⬇	Dec 1, 2006 04:55:53 PM	Laptop GPS navigation aid

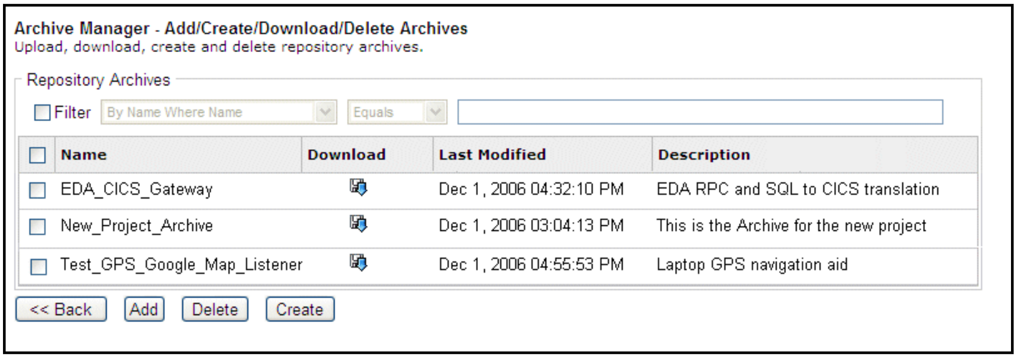
1. Select the archive file you wish to delete, and click *Delete*.

   A confirmation dialog box opens, prompting you to confirm the deletion of the archive file.

2. Click *OK* if you wish to continue the delete operation.

   Note: This deletion operation is permanent; there is no *undo* if you erroneously delete an archive, other than retrieving the archive file from the system's Recycle Bin or its original source.

As shown in the following image, the screen returns to the Archive Manager - Repository Archives page listing the *available* archive files. The Saved_SciFiBooks package is not listed as an available archive file, confirming its deletion.



Archive Manager - Add/Create/Download/Delete Archives

Upload, download, create and delete repository archives.

Repository Archives

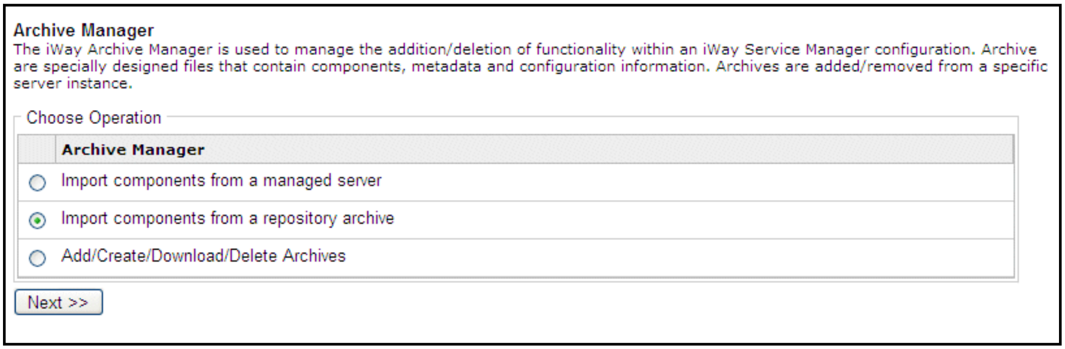
□ Filter [By Name Where Name ▾] [Equals ▾] [ ]

□	Name	Download	Last Modified	Description
□	EDA_CICS_Gateway	⬇	Dec 1, 2006 04:32:10 PM	EDA RPC and SQL to CICS translation
□	New_Project_Archive	⬇	Dec 1, 2006 03:04:13 PM	This is the Archive for the new project
□	Test_GPS_Google_Map_Listener	⬇	Dec 1, 2006 04:55:53 PM	Laptop GPS navigation aid

Importing Components From a Repository Archive

A repository archive is the primary source of components that are added to the Registry from the Archive Manager Console page.

The following image shows the Archive Manager page with the Import components from a repository archive button selected.



Archive Manager

The iWay Archive Manager is used to manage the addition/deletion of functionality within an iWay Service Manager configuration. Archive are specially designed files that contain components, metadata and configuration information. Archives are added/removed from a specific server instance.

Choose Operation

	Archive Manager
○	Import components from a managed server
●	Import components from a repository archive
○	Add/Create/Download/Delete Archives

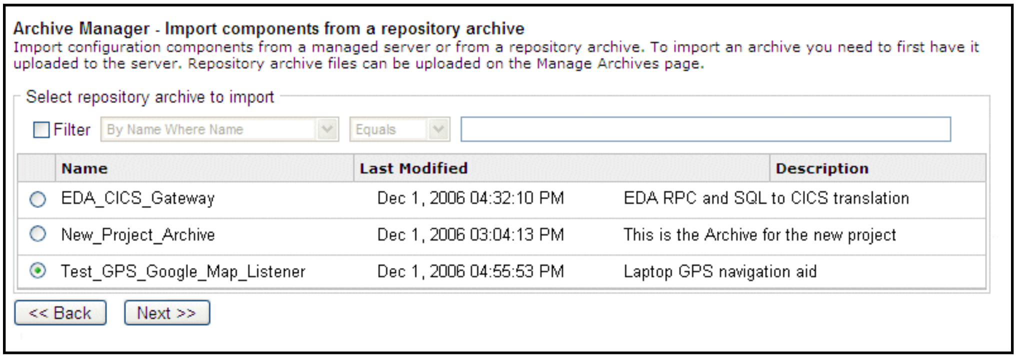
Procedure: How to Import Components From a Repository Archive

To import components from a repository archive:

1. From the left console pane, click the *Archive Manager*.

2. Select the *Import components from a repository archive* button, and click *Next*.

   The list of archive files in the [IWAYHOME]/etc/repository/manager/archives directory is displayed. You may apply filtering criteria to minimize the size of the list and help locate the archive of interest. The following image shows there is no filter being applied to the Test_GPS_Google_Map_Listener repository archive for import.



Archive Manager - **Import components from a repository archive**

Import configuration components from a managed server or from a repository archive. To import an archive you need to first have it uploaded to the server. Repository archive files can be uploaded on the Manage Archives page.

Select repository archive to import

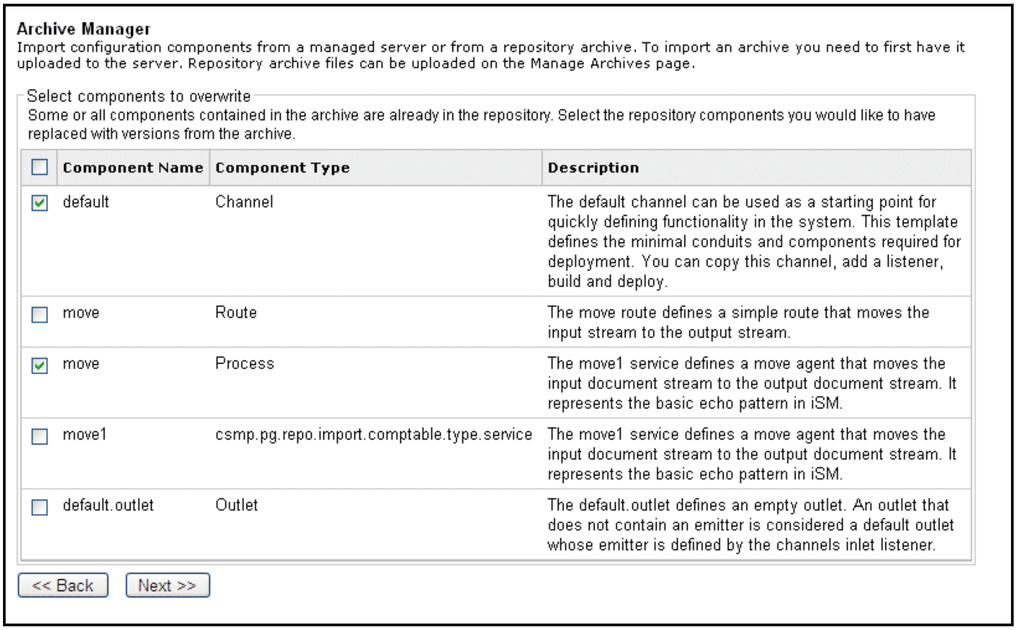
□ Filter | By Name Where Name | Equals

	Name	Last Modified	Description
○	EDA_CICS_Gateway	Dec 1, 2006 04:32:10 PM	EDA RPC and SQL to CICS translation
○	New_Project_Archive	Dec 1, 2006 03:04:13 PM	This is the Archive for the new project
●	Test_GPS_Google_Map_Listener	Dec 1, 2006 04:55:53 PM	Laptop GPS navigation aid

Note: On non-Windows platforms, to use Archive Manager, you can either upload the archive from a local machine or place the archive in the [IWAYHOME]/etc/repository/manager/archives directory. When you refresh the browser, the archive is added to the list in the Archive Manager - Import components from a repository archive page.

3. Select the archive file you wish to import, and click *Next*.

   If the components stored in the archive are already present in your Registry, you will be presented with a page to selectively replace any preexisting components. As shown in the following image, two components are selected to be overwritten, the default and move components.



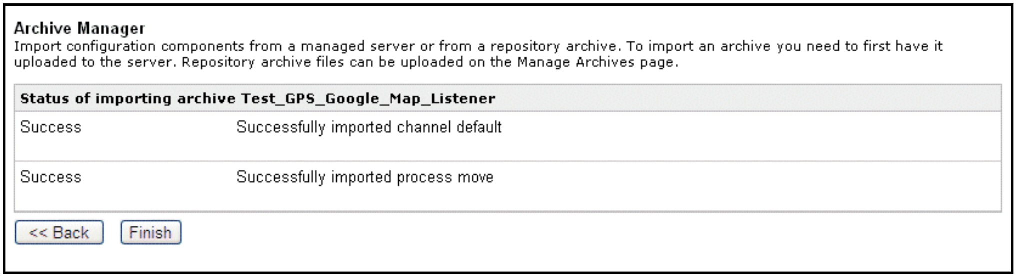
Archive Manager
Import configuration components from a managed server or from a repository archive. To import an archive you need to first have it uploaded to the server. Repository archive files can be uploaded on the Manage Archives page.

Select components to overwrite
Some or all components contained in the archive are already in the repository. Select the repository components you would like to have replaced with versions from the archive.

□	Component Name	Component Type	Description
☑	default	Channel	The default channel can be used as a starting point for quickly defining functionality in the system. This template defines the minimal conduits and components required for deployment. You can copy this channel, add a listener, build and deploy.
□	move	Route	The move route defines a simple route that moves the input stream to the output stream.
☑	move	Process	The move1 service defines a move agent that moves the input document stream to the output document stream. It represents the basic echo pattern in iSM.
□	move1	csmp.pg.repo.import.comptable.type.service	The move1 service defines a move agent that moves the input document stream to the output document stream. It represents the basic echo pattern in iSM.
□	default.outlet	Outlet	The default.outlet defines an empty outlet. An outlet that does not contain an emitter is considered a default outlet whose emitter is defined by the channels inlet listener.

4. Select the component you wish to overwrite, and click *Next* to go to the results page.

   The results page displays the success of importing the Test_GPS_Google_Map_Listener repository archive with the default and move components overwritten, as shown in the following image.



Archive Manager
Import configuration components from a managed server or from a repository archive. To import an archive you need to first have it uploaded to the server. Repository archive files can be uploaded on the Manage Archives page.

Status of importing archive Test_GPS_Google_Map_Listener	
Success	Successfully imported channel default
Success	Successfully imported process move

5. Click *Finish* to return to the main Archive Manager page.

Importing Components From a Managed Server

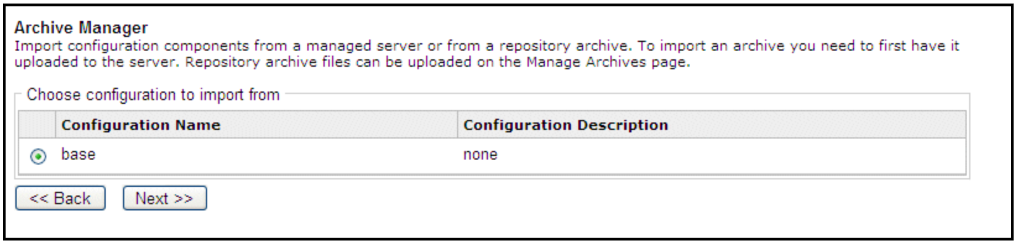
Importing components from a managed server is useful when you are required to migrate from an earlier release of iWay Service Manager. The Registry, as configured during installation, contains the components for iWay samples only. Generally speaking, you will want to insert additional components into the Registry in preparation for constructing your channels. One source of additional components is one of your managed servers (also called *configurations*).

During an upgrade, you can also import components from the run-time server into the Registry to make them available for modification.

Procedure: How to Import Components From a Managed Server

1. From the left console pane, click *Archive Manager*.
2. Select the *Import components from a managed server* button and click *Next*.

   As shown in the following image, the base (default) server is the configuration source for importing components.



Archive Manager

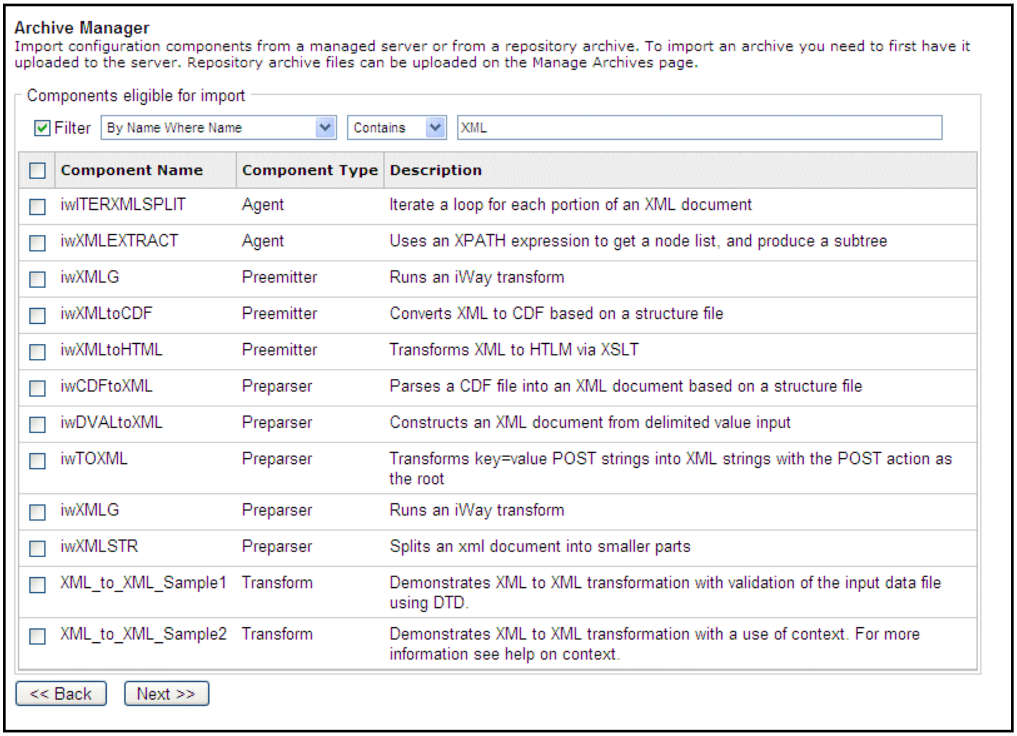
Import configuration components from a managed server or from a repository archive. To import an archive you need to first have it uploaded to the server. Repository archive files can be uploaded on the Manage Archives page.

Choose configuration to import from

	Configuration Name	Configuration Description
●	base	none

3. Click the source configuration which will be the source for your components. and click *Next*.

   All the components within the selected configuration, filtered by the selection criteria, are listed. The base configuration contains approximately 100 components, so for this example the list was limited to components whose name contains the string XML, as shown in the following image. After applying a package, such as HIPAA, more components will be available for import into the Registry.



Archive Manager

Import configuration components from a managed server or from a repository archive. To import an archive you need to first have it uploaded to the server. Repository archive files can be uploaded on the Manage Archives page.

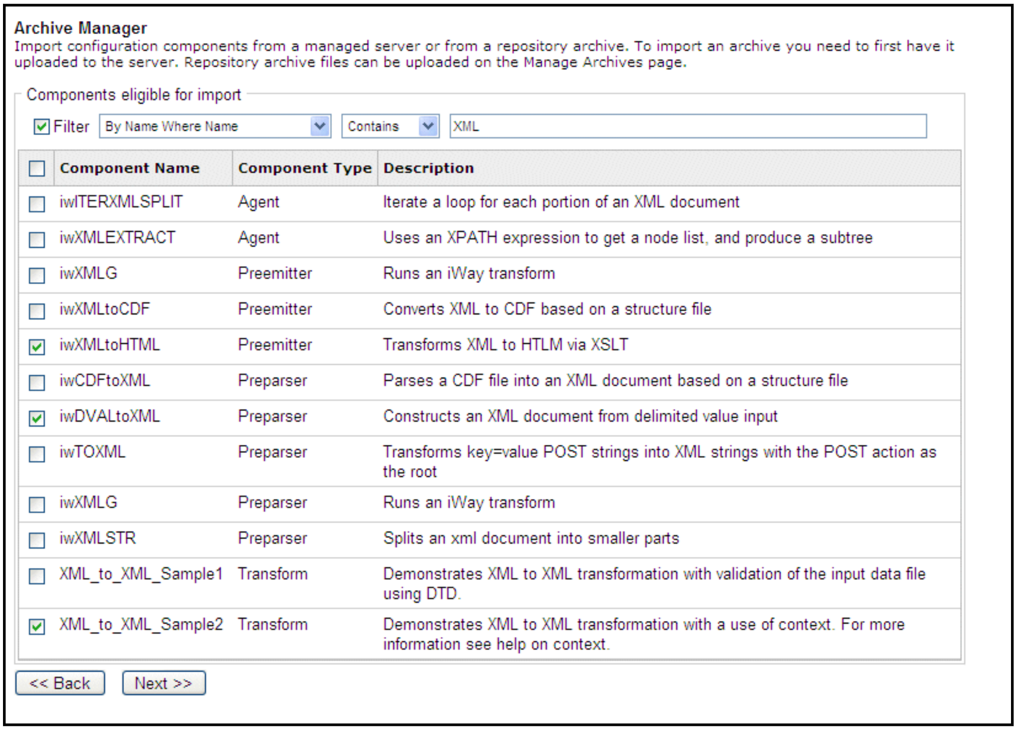
Components eligible for import

☑ Filter: By Name Where Name · Contains · XML

□	Component Name	Component Type	Description
□	iwITERXMLSPLIT	Agent	Iterate a loop for each portion of an XML document
□	iwXMLEXTRACT	Agent	Uses an XPATH expression to get a node list, and produce a subtree
□	iwXMLG	Preemitter	Runs an iWay transform
□	iwXMLtoCDF	Preemitter	Converts XML to CDF based on a structure file
□	iwXMLtoHTML	Preemitter	Transforms XML to HTLM via XSLT
□	iwCDFtoXML	Preparser	Parses a CDF file into an XML document based on a structure file
□	iwDVALtoXML	Preparser	Constructs an XML document from delimited value input
□	iwTOXML	Preparser	Transforms key=value POST strings into XML strings with the POST action as the root
□	iwXMLG	Preparser	Runs an iWay transform
□	iwXMLSTR	Preparser	Splits an xml document into smaller parts
□	XML_to_XML_Sample1	Transform	Demonstrates XML to XML transformation with validation of the input data file using DTD.
□	XML_to_XML_Sample2	Transform	Demonstrates XML to XML transformation with a use of context. For more information see help on context.

4. Select the component you wish to copy to the Registry.

   The following image shows the selection of three components to be copied to the Registry, iwXMLtoHTML, iwDVALtoXML, and XML_to_XML_Sample2.



Archive Manager

Import configuration components from a managed server or from a repository archive. To import an archive you need to first have it uploaded to the server. Repository archive files can be uploaded on the Manage Archives page.

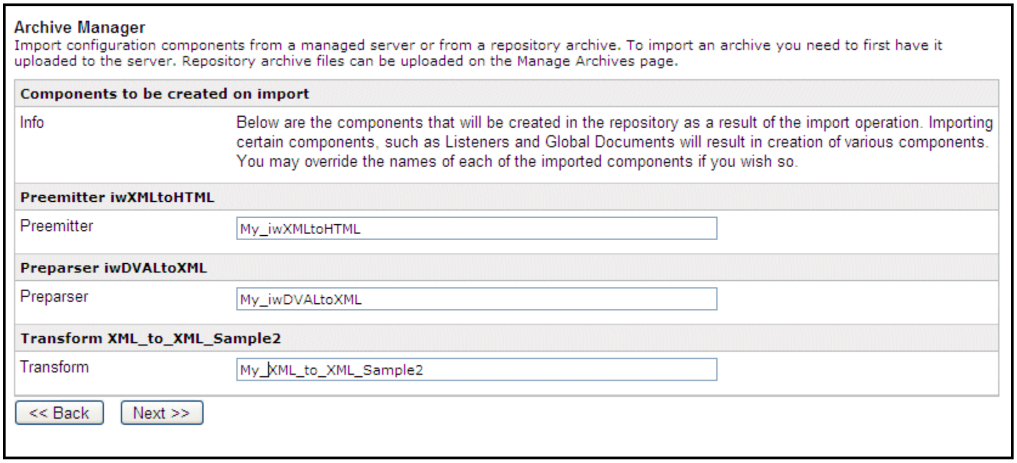
Components eligible for import

☑ Filter: By Name Where Name | Contains | XML

□	Component Name	Component Type	Description
□	iwITERXMLSPLIT	Agent	Iterate a loop for each portion of an XML document
□	iwXMLEXTRACT	Agent	Uses an XPATH expression to get a node list, and produce a subtree
□	iwXMLG	Preemitter	Runs an iWay transform
□	iwXMLtoCDF	Preemitter	Converts XML to CDF based on a structure file
☑	iwXMLtoHTML	Preemitter	Transforms XML to HTLM via XSLT
□	iwCDFtoXML	Preparser	Parses a CDF file into an XML document based on a structure file
☑	iwDVALtoXML	Preparser	Constructs an XML document from delimited value input
□	iwTOXML	Preparser	Transforms key=value POST strings into XML strings with the POST action as the root
□	iwXMLG	Preparser	Runs an iWay transform
□	iwXMLSTR	Preparser	Splits an xml document into smaller parts
□	XML_to_XML_Sample1	Transform	Demonstrates XML to XML transformation with validation of the input data file using DTD.
☑	XML_to_XML_Sample2	Transform	Demonstrates XML to XML transformation with a use of context. For more information see help on context.

5. Click *Next*.

In the following image, this page allows you to rename the newly created Registry components. The components have the prefix My_ for identification and uniqueness.



Archive Manager

Import configuration components from a managed server or from a repository archive. To import an archive you need to first have it uploaded to the server. Repository archive files can be uploaded on the Manage Archives page.

Components to be created on import

Info	Below are the components that will be created in the repository as a result of the import operation. Importing certain components, such as Listeners and Global Documents will result in creation of various components. You may override the names of each of the imported components if you wish so.

Preemitter iwXMLtoHTML

Preemitter	My_iwXMLtoHTML

Preparser iwDVALtoXML

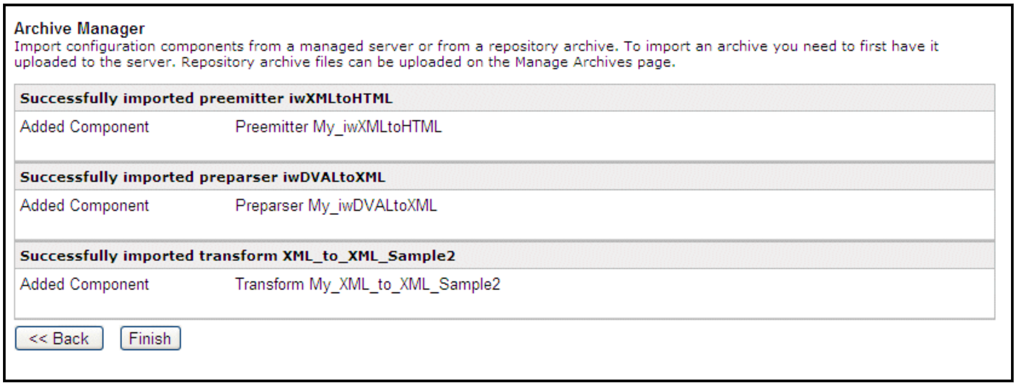
Preparser	My_iwDVALtoXML

Transform XML_to_XML_Sample2

Transform	My_XML_to_XML_Sample2

6. Click *Next* to continue to the confirmation/summary page.

   The following image shows that the preemitter, preparser, and transform components have been successfully imported.



Archive Manager

Import configuration components from a managed server or from a repository archive. To import an archive you need to first have it uploaded to the server. Repository archive files can be uploaded on the Manage Archives page.

Successfully imported preemitter iwXMLtoHTML

Added Component	Preemitter My_iwXMLtoHTML

Successfully imported preparser iwDVALtoXML

Added Component	Preparser My_iwDVALtoXML

Successfully imported transform XML_to_XML_Sample2

Added Component	Transform My_XML_to_XML_Sample2

7. Click *Finish* to return to the main Archive Manager page.

Using the Deployment Manager

iWay Service Manager provides support for servlet deployment within various application servers, such as Apache Tomcat, IBM WebSphere, Sun Application Server, and JBoss. The Deployment Manager allows you to create a standalone configuration under iWay Service Manager and then create a corresponding web application (.WAR) that encapsulates the configuration. The web application includes all the required components for the configuration and the iWay Software components necessary to run iWay Service Manager as a standalone web application.

Note: Any required third-party components must be registered separately with the application server. The deployed iWay Service Manager servlet is designed to be used as a run-time environment. Only one .WAR can be used per Application Server configuration.

Deploying iWay Service Manager as a Servlet

You must first install the iWay Service Manager release on your local system and then create a configuration that can include channels, web services, and any other required components for the solution. Once the configuration has been fully tested and is ready to be deployed to the Application Server, follow the steps below.

Procedure: How to Create Web Applications Based on iWay Service Manager Configurations

To create web applications based on iSM configurations:

1. Open a web browser and point to the following URL:

   ```
   http://host:port
   ```

   where:

   `host`

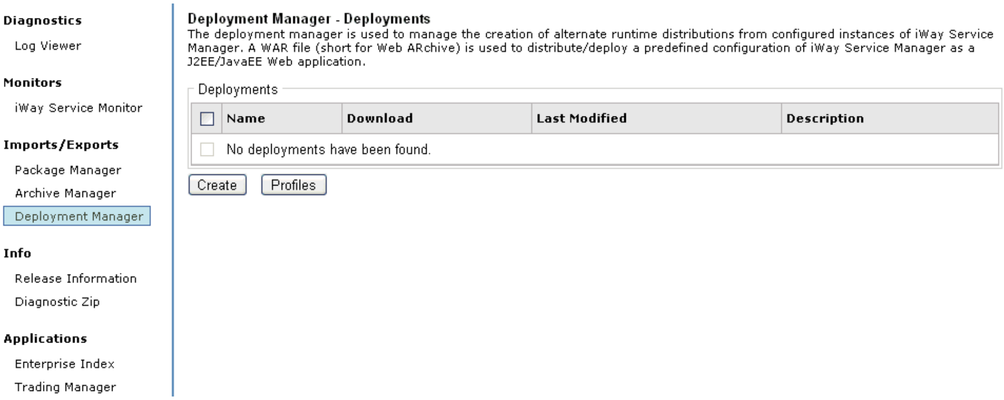
   Is the name of the server where iSM is installed.

   `port`

   Is the port on which the server console is listening. The default is 9999.

2. In the top pane, click *Tools*.
3. From the Imports/Exports list on the left pane, click *Deployment Manager*.

The Deployment Manager pane opens.



Deployment Manager - Deployments

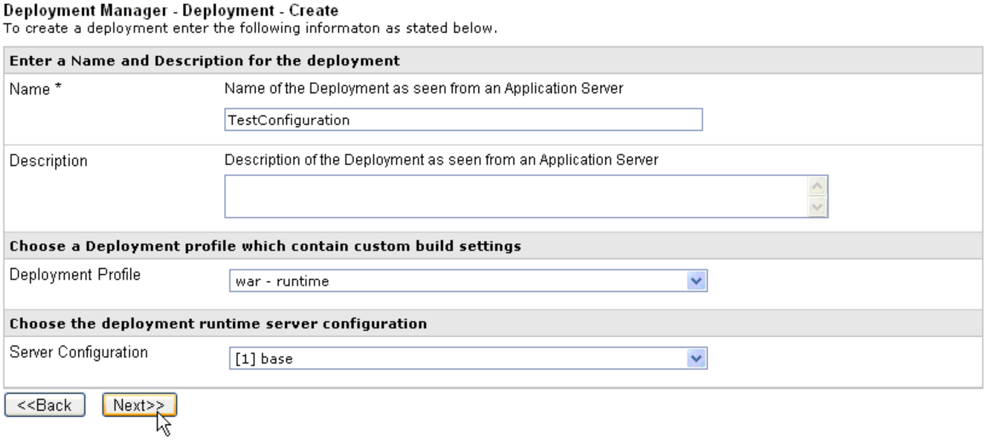
The deployment manager is used to manage the creation of alternate runtime distributions from configured instances of iWay Service Manager. A WAR file (short for Web ARchive) is used to distribute/deploy a predefined configuration of iWay Service Manager as a J2EE/JavaEE Web application.

Deployments

□	Name	Download	Last Modified	Description
□	No deployments have been found.			

4. Click *Create* to add a new web application module.

5. Provide the required configuration parameters, and click *Next*.



Deployment Manager - Deployment - Create

To create a deployment enter the following informaton as stated below.

Enter a Name and Description for the deployment

Name * — Name of the Deployment as seen from an Application Server: TestConfiguration

Description — Description of the Deployment as seen from an Application Server

Choose a Deployment profile which contain custom build settings

Deployment Profile: war - runtime

Choose the deployment runtime server configuration

Server Configuration: [1] base

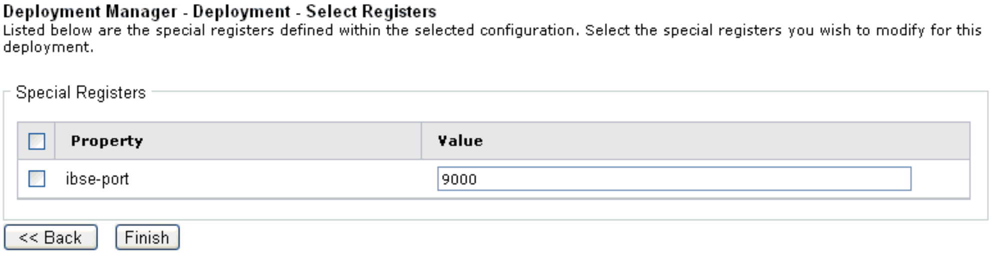
The following table lists and describes each parameter for the Deployment Manager.

Parameter Name	Description
Name	Name of the web application module.
Descriptions	Optional description for the created module.

Parameter Name	Description
Deployment Profile	Lists the available deployment profiles: ❑ *war - runtime:* Includes the required components for the runtime. ❑ *war - runtime - adapter:* Includes the required components for the runtime and the adapter configurations. Additional custom profiles may be created.
Server Configuration	List of available server configurations. Select the configuration for which you created a deployment. You do not need to select base since it is included by default.

The Special Registers pane opens, which list the special registers defined within the selected configuration. Select the special registers you wish to modify for this deployment.

Note: Do not select the ibse port. The default application server port will be used for running web services.



Deployment Manager - Deployment - Select Registers

Listed below are the special registers defined within the selected configuration. Select the special registers you wish to modify for this deployment.

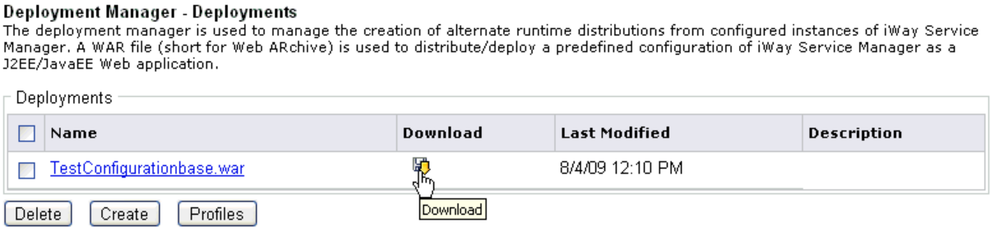
Special Registers

□	Property	Value
□	ibse-port	9000

6. Click *Finish*.

   After the process is complete, the Deployment file is displayed in a list and can be selected for download.

To download the module click on the download icon as shown in the following image.



Deployment Manager - Deployments

The deployment manager is used to manage the creation of alternate runtime distributions from configured instances of iWay Service Manager. A WAR file (short for Web ARchive) is used to distribute/deploy a predefined configuration of iWay Service Manager as a J2EE/JavaEE Web application.

Deployments

□	Name	Download	Last Modified	Description
□	TestConfigurationbase.war		8/4/09 12:10 PM	

The downloaded Deployment file can now be deployed to an Application Server and will reflect the contents of the deployed iSM configuration that was selected.

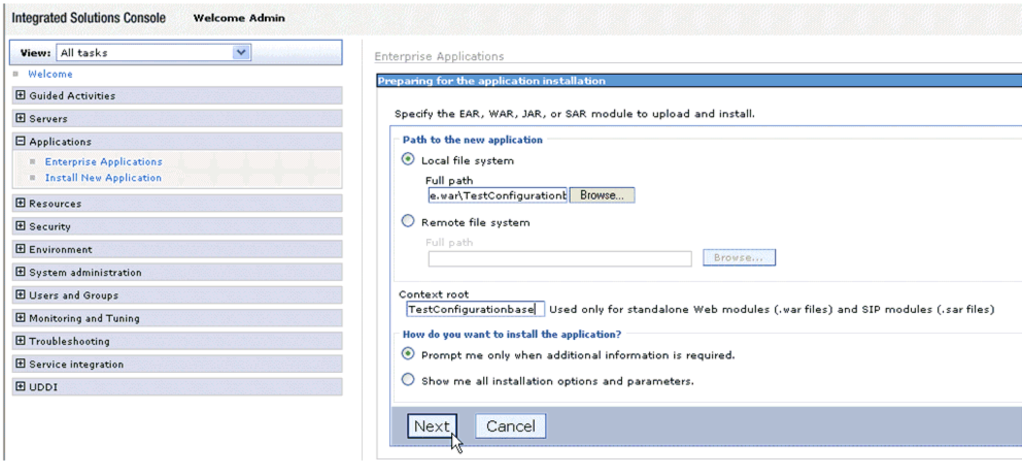
Deploying Web Application Module to WebSphere

This section describes the deployment of the iSM-based web application module to the WebSphere environment. This process would be different depending on the Application Server. Refer to the User Guide of your Application Server for details on the Application Deployment. To deploy within the WebSphere 6.1 environment you can follow the steps provided.

Procedure: How to Deploy the Web Application Module to WebSphere

To deploy the web application module to WebSphere:

1. Connect and log on to the WebSphere Administrative Console.
2. In the left panel, expand the *Applications* tab and select *Install New Application*.



Specify the EAR, WAR, JAR, or SAR module to upload and install.

Path to the new application

● Local file system

Full path: e.war\TestConfigurationb Browse...

○ Remote file system

Full path: Browse...

Context root: TestConfigurationbase Used only for standalone Web modules (.war files) and SIP modules (.sar files)

How do you want to install the application?

● Prompt me only when additional information is required.

○ Show me all installation options and parameters.

3. Provide the location of the created iSM-based war archive along with the *Context root* and continue with the installation by clicking *Next*.

Note: The *Context root* must be consistent with the web.xml file.

On the following screens of the WebSphere deployment process, you can either accept all the default configuration parameters or modify them to fit your WebSphere environment.

4. Deploy the web application module and save the configuration to the WebSphere server.

   Once it is saved successfully, the application can be managed by navigating to *Applications* and selecting *Enterprise Applications*. It is recommended to restart the server after the installation has been completed.

5. To test the deployed iSM based Servlet, connect to the web application URL:

   `http://host:port/Contextroot/console`

   where:

   `host`

   Is the server where the web application is running.

   `port`

   Is the port where the web application is running.

   `Contextroot`

   Is the context root provided for the application.

   You will be prompted for the user/password to log on to the iSM console. After providing the information, you can use the full iSM Console to manage your configuration and ensure that it is running properly.

Understanding the WEB.XML Configuration File

The web.xml file is a text-based XML file that provides configuration and deployment information for the web components that comprise a web application. The web.xml file resides in the WEB-INF directory under the context of the hierarchy of directories that exist for a web application. Examples of web components are servlet parameters, servlet and JavaServer Pages (JSP) definitions, and Uniform Resource Locators (URL) mappings. The Java Servlet 2.4 specification defines the web.xml deployment descriptor file in terms of an XML schema document. For backwards compatibility of applications written to the Java Servlet 2.3 specification, web containers are also required to support the Java Servlet 2.3 specification.

You can deploy iWay Service Manager as a web application. Additionally, you can use the Deployment Manager to create a customizable and reusable web application.

The following sample illustrates a typical **web.xml** that is created by the Deployment Manager:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
       xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       id="homebase" version="2.4">
   <display-name>iWay 7 smsp2.13562 - homebase</display-name>
   <servlet>
      <servlet-name>homebase</servlet-name>
      <servlet-class>com.ibi.edaqm.ServiceManager</servlet-class>
      <init-param>
         <param-name>console</param-name>
         <param-value>web</param-value>
      </init-param>
      <init-param>
         <param-name>config</param-name>
         <param-value>base</param-value>
      </init-param>
      <init-parm>
         <param-name>MySpecialRegister</param-name>
         <param-value>ABC123</param-value>
      </init-parm>
   </servlet>
   <servlet-mapping>
       <servlet-name>homebase</servlet-name>
       <url-pattern>/*</url-pattern>
   </servlet-mapping>
</web-app>
```

Reserved Parameters

This section describes the group of special predefined parameters that are reserved to help control the behavior of iWay Service Manager.

❑ This parameter names the configuration that the servlet is based on. If omitted its default value is **base**.

```
<init-param>
       <param-name>config</param-name>
       <param-value>base</param-value>
</init-param>
```

❑ **console**

This optional parameter controls the support of a console. It may be omitted or its value may be left empty. If it is missing or invalid, no console support is assumed. Currently the only valid value is **web**.

```
<init-param>
       <param-name>console</param-name>
       <param-value>web</param-value>
</init-param>
```

❑ **ibsp**

This optional parameter names the location of the IBSP that the iWay adapter will reference. Note that the URL defined in the connection parameters of iWay adapter targets (typically set to `http://localhost:9000`) changes as iSM is deployed as a servlet. To accommodate this change we instruct IBSP to replace the target endpoints as follows. The value is taken as a base URL.

```
<init-param>
       <param-name>ibsp</param-name>
       <param-value>http://localhost:8080</param-value>
</init-param>
```

User Defined Parameters

In addition to the reserved parameters, there may be additional parameters in your web.xml file. The web application driver of iWay Service Manager treats these parameters as System Special Registers. This allows an Administrator the reconfigure the values of these variables by updating the web.xml file according to your specific application server implementation. For example:

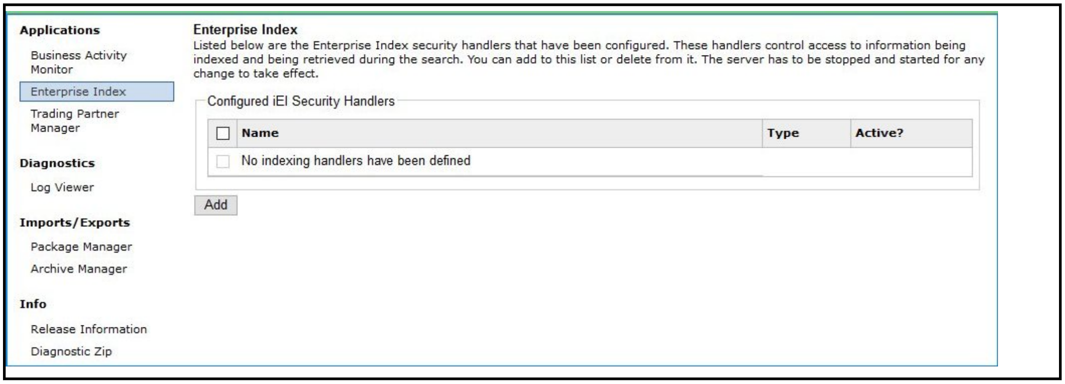
```
<init-param>
       <param-name>MySpecialRegister</param-name>
       <param-value>ABC123</param-value>
</init-param>
```

Using the Enterprise Index Application

iWay Enterprise Index (iEI) provides a secure, user-friendly way to search iWay Service Manager messages for specific content, regardless of the order or context in which the content criteria appears in the message.

iEI enables you to make Google-style Key Word Out of Context (KWOC) inquiries to all messages that pass though iSM, and provides security to prevent unauthorized parties from accessing information for which they do not have access rights.

You can add iEI access handlers using the iWay Service Manager Administration Console, which control access to information being indexed and retrieved during the search.



Enterprise Index

Listed below are the Enterprise Index security handlers that have been configured. These handlers control access to information being indexed and being retrieved during the search. You can add to this list or delete from it. The server has to be stopped and started for any change to take effect.

Configured iEI Security Handlers

□	Name	Type	Active?
□	No indexing handlers have been defined		

You can add or delete iEI access handlers as required. You must restart iWay Service Manager for any changes to take effect. For more information on using iEI, see the *iWay Enterprise Index User’s Guide*.

Chapter 10

Operations and Monitoring

This section describes basic operational, monitoring, testing, and management features of iWay Service Manager (iSM).

In this chapter:

Understanding iWay Service Manager Restart Methods

When making changes using iWay Service Manager that impact your Java Runtime Environment (JRE), the CLASSPATH variable, or special register settlings, iWay Service Manager must be restarted to apply these changes. The following methods of restarting iWay Service Manager are available:

❑ Full Restart

❑ Console Restart

By understanding the differences between these two methods, you can determine which restart method is required depending on the type of change that is made.

A full restart, which is also known as a cold or hard restart, must be used whenever changes to the JRE and CLASSPATH variable are made. These also include changing Java Virtual Machine (JVM) options, register libraries, providers, and adding files to the iway7/lib directory. In the case of a full restart, all portions of the JRE are loaded from disk.

A console restart, which is also known as a warm or soft restart, is faster than a full restart and can be used when changes to special register settings are made. In the case of a console restart, most of the JRE is already available in the cache, and is loaded from RAM instead of from disk.

For more information on restarting iWay Service Manager, see *Class Path and Service Manager Restart Issues* on page 400.

Using iWay Service Manager Startup Command Line Switches

You can use various command line switches to start iWay Service Manager (iSM) for special processing control. To view a list of all available switches use the -> or *-help* command line options.

If you are using the standard provided startup scripts, you must specify an iSM configuration to add the switches. For example:

```
iway7 base -help
```

Note: To reduce the probability of switch clashing on the iSM startup command line (since iwsrv and iSM use the same code), any switch can also be started with capital I. For example, using *-Ib* as a command line switch to start iSM in hot backup mode is identical to *-b*. Specifying the initial letter is optional.

When entering a switch with a value (such as *-srvr.lpath*), you can separate the value by a space or an equals sign (“=”). When an equals sign is used, spaces are not accepted.

Switches should be entered on the *Java Settings* page of the command console.

Key startup command line switches are listed and described in the following table.

Startup Command Line Switch	Description
-config *<name>*	Name of a specific iSM configuration or iWay Integration Application (iIA) to start, which is required to start iSM.

Startup Command Line Switch	Description
`-b[ackup]`	Starts iSM in hot backup mode. If your iSM instance shadows another iSM instance for backup purposes (as configured in the Backup Settings section of the iSM Administration Console), using the *-b* startup command line switch starts iSM as the shadowing (backup) server. This switch does not apply to backup situations where a backup listener controls the backup process.
`-n[ame]` *<name>*	Sets the name that is assigned to iSM. Usually this is the host or configuration name, but can be modified by this switch. The use of this startup command line switch is reserved for future use.
`-v[ersion]`	Displays current version information for your iSM instance.
`-r[aw]`	Bypasses the use of a startup process flow. iSM will start without attempting to run the configured startup process flow.
`-u[sersecurity]`	Ignores the iSM Authentication Realm settings and uses the iSM Administration Console realm instead. You can use this startup command line switch in a scenario where you are required to adjust the realm information and cannot access the iSM Administration Console to do so because a realm (such as LDAP) is unreachable.
`-srvr.quiesced`	Quiesced mode, all channels except the console channel are inactive on startup.
`-srvr.ifile` `<path>`	Path to install a file (usually install.xml). Defaults to: `[iwayhome]/bin`
`-srvr.lpath` `<path>`	Path to the directory where license file(s) are located. Defaults to: `[iwayhome]`

The path setting can be used to modify the Java classpath and native path libraries that are used. These settings apply to Windows and Unix startup commands. For more information on iSM path settings, see *Path Settings* on page 130.

Additional startup command line switches are used internally by iSM, and are not recommended for use in production environments.

Starting and Stopping iWay Service Manager on Windows

If you are *not* running on Windows, see *Starting and Stopping Service Manager on UNIX, OS/400, and z/OS* on page 395.

You can run Service Manager on Windows in one of the following ways:

- ❑ Service Manager can run in the background as a Windows service.

  Use this method in most situations. For more information, see *Starting and Stopping Service Manager as a Windows Service* on page 392.

- ❑ Service Manager can run in a command window for debugging and troubleshooting.

  For more information, see *Starting and Stopping Service Manager in the Command Window* on page 394.

On Windows, the startup procedure uses the configuration files as follows:

1. The iwsrv executable program is run as a service or manually, for example:

   ```
   C:\Program Files\iway8\bin\iwsrv.exe
   ```

2. When Service Manager starts as a service or in a command window, it locates the required JAR files and the files you successfully registered.
3. Service Manager loads the Java Runtime Environment (JRE) by looking for versions and locations in the Windows registry. Java version 1.6 is required.

Starting and Stopping Service Manager as a Windows Service

When Service Manager is installed, a Windows service for the “base” configuration is created. By default, this service starts with Windows whenever you reboot the system. You can manually start and stop the service as explained in the following procedures.

When running Service Manager as a Windows service, Service Manager runs in the background as the Local System account and remains running if you log off the system. For assistance with troubleshooting, check the Event Viewer as explained in *Diagnostics, Tracing, and Logging* on page 471.

You can start the Service Manager service from either the Windows Services window or through the Start menu. You also can stop the Service Manager service from the Windows Services window or through the Start menu.

By default, the Service Manager service runs as the Local System account and starts automatically when you reboot the Windows system. If you do not want Service Manager to start automatically with Windows, or you want to run Service Manager with a different user ID, see *How to Change the Service Manager Windows Service Configuration* on page 393.

Tip: To create Windows services for other configurations, see *Managing Configurations* on page 89.

Procedure: How to Start or Stop Service Manager From the Windows Services Window

To use the Services window to start or stop Service Manager as a Windows service:

1. From the Control Panel, select *Administrative Tools*, and then *Services*.

   The Services window opens.

2. To start Service Manager, right-click *iWay Service Manager - base* and select *Start*.

   The service status changes to started.

3. To stop Service Manager, right-click *iWay Service Manager - base* and select *Stop*.

Procedure: How to Start or Stop Service Manager From the Windows Start Menu

To use the Windows Start menu to start or stop Service Manager as a Windows service:

1. Select *Start*, *Programs*, and then *iway8 Service Manager*.
2. To start Service Manager, select *base*, and then *Start Service Manager - base*.
3. To stop Service Manager, select *base*, and then *Stop Service Manager - base*.

Procedure: How to Change the Service Manager Windows Service Configuration

To prevent Service Manager from starting automatically or to run Service Manager with a user ID other than Local System:

1. From the Control Panel, select *Administrative Tools*, and then *Services*.

   The Services window opens.

2. In the list, double-click *iWay Service Manager - base*.

   The iWay Service Manager-base Properties dialog box opens.

   a. To prevent Service Manager from starting automatically with Windows, change the Startup type to *Manual*.

   b. Click *Apply*.

3. To run Service Manager with a user ID other than Local System:

   a. Click the *Log On* tab.

   b. Select the *This account* button.

   c. Specify the user ID and password that you want to run Service Manager.

      The user ID must be an administrator of the local system.

      If you change the user ID under which Service Manager runs, ensure that the user ID has full NTFS permissions to the iway8 directories.

4. Click *OK* to close the dialog box.

Starting and Stopping Service Manager in the Command Window

For debugging purposes, you can run Service Manager in a command window instead of running it as a Windows service.

The command window enables you to interact with Service Manager and displays diagnostic information about Service Manager processing.

Procedure: How to Start Service Manager in the Command Window

To start Service Manager in the command window:

1. Open a command window and navigate to the iway8\bin directory, for example:

   ```
   C:\Program Files\iway8\bin
   ```

2. At the command prompt, type the following command to start the base configuration:

   ```
   iwsrv
   ```

   Note: For other configurations, enter the configuration name after iwsrv. Configuration names are case-sensitive.

   Messages appear in the command window detailing the startup process. The following prompt appears when the start up is complete:

   ```
   Enter command:>
   ```

   If you receive errors similar to the following, Service Manager is probably already running as a service. You cannot run the same configuration at the same time as both a service and in the command window.

```
ERROR (HTTP1) Unable to create server socket 9980:
java.net.BindException: Address already in use: JVM_Bind
ERROR (SOAP1) Unable to create server socket 9000:
java.net.BindException: Address already in use: JVM_Bind
```

Procedure: How to Stop Service Manager in the Command Window

To stop Service Manager in the command window:

1. Go to the following Service Manager prompt:

   ```
   Enter command:>
   ```

2. At the Service Manager prompt, type the following:

   ```
   stop
   ```

3. When any listeners that are running have stopped, type the following:

   ```
   quit
   ```

Note: If you close the command window without typing these commands, Service Manager stops. However, typing the commands is recommended.

Starting and Stopping Service Manager on UNIX, OS/400, and z/OS

If you are running on Windows, go to *Class Path and Service Manager Restart Issues* on page 400.

Shell scripts are installed with iWay to run Service Manager on UNIX, OS/400, and z/OS systems. The shell scripts require that the JAVA command is included in your $PATH environment variable. Environment variables must be set for the user IDs that start Service Manager as explained in the *iWay Installation and Configuration* documentation.

The following are ways to run Service Manager:

❑ Service Manager can run in the background as a service (daemon). For more information, see *Starting and Stopping Service Manager as a Service (Daemon)* on page 396.

❑ Service Manager can run in the shell to display diagnostic information and an interactive prompt. For more information, see *Starting and Stopping Service Manager in the Shell* on page 398.

❑ On z/OS systems, Service Manager can be started in batch mode using JCL. For more information, see *Starting and Stopping Service Manager as a Batch Process* on page 399.

Note: Running Service Manager normally refers to running the base configuration unless another configuration is specifically mentioned. When you installed Service Manager, a base configuration of Service Manager was created and this is used by default. Also note the following:

- ❏ UNIX instructions apply to Linux.
- ❏ On OS/400, the following procedures should be performed in QSH.
- ❏ z/OS instructions apply to OS/390.
- ❏ On z/OS, the non-JCL procedures should be performed in MVS.

Starting and Stopping Service Manager as a Service (Daemon)

When you run Service Manager as a service (daemon), it runs in the background as the user ID specified in its startup and shutdown scripts. Shell scripts to start and stop Service Manager as a service (daemon) are in the iway7/bin directory.

On UNIX, z/OS, and OS/400, the startup script is:

```
iway7/bin/startservice.sh
```

On UNIX, z/OS, and OS/400, the shutdown script is:

```
iway7/bin/stopservice.sh
```

Note: On z/OS, these scripts must be EBCDIC.

The scripts set several environment variables. On UNIX and OS/400, the installation program sets them for you. You can edit the scripts if you wish to change these settings. On z/OS, you must manually edit the scripts to set the following variables.

`IWAY7SM`

Is the location where iWay Service Manager is installed.

`IWAYUSER`

Is the user ID under which Service Manager runs. When you run the scripts as a non-super user ID (not root), you are prompted for the user ID password. On most UNIX systems, the default is root. For security reasons, you may decide not to run Service Manager as root.

`IWAYCONFIG`

Is the configuration to run. This is set to base by default.

The scripts also specify the log file for the service. By default these are:

```
iway7/serviceOut.txt
```

```
iway7/serviceShutdown.txt
```

If you create copies of the startup and shutdown scripts in order to run multiple configurations, ensure each script has a unique log file.

Note: If the JAVA command is in your $PATH variable, you usually are not required to set it in the script. However, if you run into problems, edit the script so that the full path to the JAVA command appears instead of "java."

Procedure: How to Start Service Manager as a Service (Daemon)

To start Service Manager as a service (daemon):

1. Navigate to the iway7/bin directory, for example:

   ```
   /opt/iway7/bin
   ```

2. Execute the service startup script file, for example:

   ```
   ./startservice.sh
   ```

   If you are not a super user, you are prompted for the password of the user ID under which Service Manager runs.

   ```
   Password:
   ```

3. Type the password for this user ID.

   The configuration of Service Manager is started in the background and includes SOAP and HTTP listeners. A serviceOut.txt file appears in the iway7 directory and contains log information.

Procedure: How to Stop Service Manager as a Service (Daemon)

To stop Service Manager as a service (daemon):

1. Navigate to the iway7/bin directory, for example:

   ```
   /opt/iway7/bin
   ```

2. Execute the service shutdown script file, for example:

   ```
   ./stopservice.sh
   ```

   If you are not a super user, you are prompted for the password of the user ID under which Service Manager runs.

   ```
   Password:
   ```

3. Enter the password for this user ID.

   Note: If you receive an error, ensure the user ID is defined in the shutdown script file.

   A serviceShutdown.txt file appears in the iway7 directory and contains log information about the shutdown process.

Starting and Stopping Service Manager in the Shell

For debugging purposes, you can run Service Manager in the shell (or QSH) in an interactive and diagnostic (non-service) mode.

Procedure: How to Start Service Manager in the Shell

To run Service Manager in the shell as a non-service:

1. Navigate to the iway7 directory, for example:

   ```
   /opt/iway7
   ```

2. Start Service Manager by typing the following command:

   ```
   ./iway7.sh configname
   ```

   where:

   configname

   Is the name of the Service Manager configuration you want to start.

   Service Manager starts and messages display information about the startup process. When Service Manager has completely started, the following prompt appears:

   ```
   Enter command:>
   ```

   This prompt enables you to interact with Service Manager and displays diagnostic information about Service Manager processing.

   Caution: If you previously ran Service Manager as a service under a user ID such as root and then run Service Manager in the shell as a different user ID, you may encounter permission problems. Files created while Service Manager ran as a service may not be readable or writable to the user ID running Service Manager in the shell.

Procedure: How to Stop Service Manager in the Shell

To stop Service Manager running in the shell as a non-service:

1. Go to the following Service Manager prompt:

   ```
   Enter command:>
   ```

2. At the Service Manager prompt, type the following to stop listeners:

   ```
   stop
   ```

3. When listeners have stopped, type the following:

   ```
   quit
   ```

Starting and Stopping Service Manager as a Batch Process

On non-z/OS systems, go to *Class Path and Service Manager Restart Issues* on page 400.

On z/OS systems, you can also start Service Manager as a batch process using JCL.

Example: Starting Service Manager as a Batch Process on z/OS

The following sample JCL starts Service Manager in batch. Substitute the appropriate job card and HFS locations on your system.

```
//EDABGBPX JOB (SMITH),'JAVA BPXBATCH',CLASS=A,MSGLEVEL=(1,1),
//   MSGCLASS=X,REGION=0M,NOTIFY=EDABG,USER=EDABG1,PASSWORD=XXXXXXX
//********************************************************************
//* RUN JAVA UNDER A UNIX SYSTEM SERVICE SHELL
//********************************************************************
//STEP2 EXEC PGM=BPXBATSL,
// PARM='PGM /bin/sh /u/edabg1/iway7sm/iway7.sh base -c'
//STDOUT DD SYSOUT=*
//STDERR DD SYSOUT=*
//STDENV DD *
JAVA_HOME=/usr/lpp/java/J1.5
PATH=/usr/lpp/java/J1.5/bin
//
```

Example: Stopping Service Manager as a Batch Process on z/OS

The following sample JCL stops Service Manager in batch. Substitute the appropriate job card and HFS locations on your system.

```
//EDABGBPS JOB (SMITH),'JAVA BPXBATCH',CLASS=A,MSGLEVEL=(1,1),
//   MSGCLASS=X,REGION=0M,NOTIFY=EDABG,USER=EDABG1,PASSWORD=XXXXXXX
//********************************************************************
//* RUN JAVA UNDER A UNIX SYSTEM SERVICE SHELL
//********************************************************************
//STEP2 EXEC PGM=BPXBATSL,
// PARM='PGM /bin/sh /u/edabg1/iway7sm/bin/iway7sd.sh'
//STDOUT DD SYSOUT=*
//STDERR DD SYSOUT=*
//STDENV DD *
PATH="/usr/local/diff/bin:.:/usr/lpp/java/J1.5
      /bin:/bin::/usr/local/bin /usr/local/subin:/usr/sbin"1.5/bin'
LIBPATH=/lib:/usr/lib:/usr/lpp/java/J1.5/bin:.
JAVA_HOME=/usr/lpp/java/J1.5
//
```

Class Path and Service Manager Restart Issues

You should be aware of the information in the following topics about setting the class path and restarting Service Manager.

Class Path

When Service Manager starts, it loads the Java VM and adds all JAR or ZIP files in the iway7/lib directory to its CLASSPATH variable. To include files outside of iway7/lib in class path, register them as explained in *Configuring General Properties Using the Console* on page 111. To exclude files in iway7/lib from being in class path, move the files to another directory or rename them so that they do not end in *.jar* or *.zip* (a file with an extension such as .jarold would be included).

When you access the Service Manager console in a web browser, the first window displays a class path drop-down list containing all files in the class path. When you register a library or add files to the CLASSPATH variable, you should ensure that the files appear in this drop-down list after you stop and then restart Service Manager.

Trace Level Settings

Trace level settings (for example, debug, deep, external) can be changed using the console or the command line. When setting trace levels, it is not necessary to restart your iSM instance. Trace level settings are applied automatically to the running iSM instance.

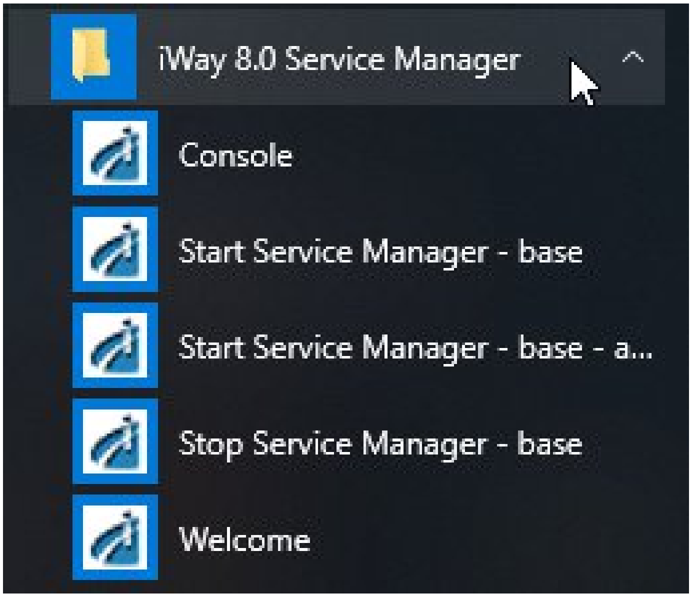
Restarting Service Manager

You can restart Service Manager using one of the following methods:

- **Full Restart**

A full restart (stopping and starting Service Manager) ensures all changes are applied and a completely new Java Runtime Environment (JRE) is used when Service Manager restarts. This is required for any changes that affect the JRE. You must perform a full restart if you change register libraries, providers, and Java Virtual Machine (JVM) options. In addition, a full restart is also required if you add new files to the iway7/lib directory or add files to the CLASSPATH variable.

The following image shows the Start and Stop options for Service Manager running on Windows.

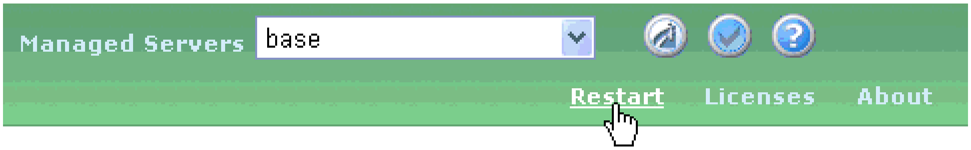

iWay 8.0 Service Manager
Console
Start Service Manager - base
Start Service Manager - base - a...
Stop Service Manager - base
Welcome

For more information, refer to the procedures appropriate for your platform and the mode in which Service Manager runs (for example, service or non-service).

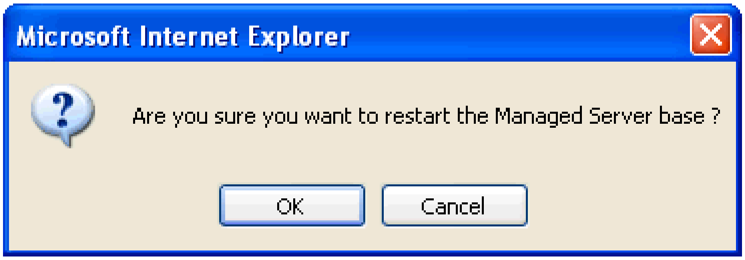
- ❑ **Console Restart**

  A console restart ensures that changes made to special register settings are applied. It does not apply any system level changes, as described by the full restart method.

  The following image shows the Restart link in Service Manager that is used to restart the iWay Service Manager Administration Console.



When you click *Restart*, the following dialog is displayed:



Are you sure you want to restart the Managed Server base ?

Click *OK* to restart. After the console restarts, you are returned to the General Properties page in the iWay Service Manager Administration Console.

Understanding and Using the IWSRV Executable Program

The *iwsrv* executable program allows you to install a Windows service for iWay Service Manager (iSM) that can be used to start an iSM configuration. You can also start iSM using *iwsrv* in a command window for debugging purposes.

IWSRV Options

This section lists and describes all of the available options for the *iwsrv* executable program.

The full syntax for the *iwsrv* command is:

```
iwsrv [config_name] [-s service] [-l launch] [options]
```

where:

config_name

Is the name of the iSM configuration that is loaded for this instance. The default iSM configuration is *base*.

service

Is the name of the iSM Windows service that is executed. Valid values are:

start: Starts the iSM configuration for the Windows service (default). For more information, see *How to Start iSM in a Command Window* on page 405.

stop: Stops the iSM configuration for the Windows service. For more information, see *How to Stop iSM in a Command Window* on page 406

install: Installs the iSM Windows service. For more information, see *How to Install an iSM Windows Service* on page 407.

remove: Deletes the iSM Windows service. For more information, see *How to Delete an iSM Windows Service* on page 407.

query: Displays the options that were used to install an iSM Windows service. For more information, see *How to Query the Install Options for an iSM Windows Service* on page 410.

launch

Specifies the launch method. Valid values are:

java: Loads Java in a separate process and uses the JVM options, NT dependencies, and other preferences found within the iSM configuration that are configured through the console.

For example:

```
iwsrv config_name -s start -l java
```

where:

config_name

Is the name of the iSM configuration.

Using the -l option forces the iSM Windows service to load Java in a separate process. When the iSM Windows service is stopped, iwsrv.exe and java.exe are terminated.

options

Specifies tracing or server backup information. Valid values include:

-b: Indicates that iSM is a backup server.

For example:

```
iwsrv config_name -s start -b
```

where:

config_name

Is the name of the iSM configuration.

-c: Enables tracing. In this mode, you can display useful error messages on the console. For example, you can display a message which indicates that the Java Runtime Environment (JRE) is not properly installed.

For example:

```
iwsrv config_name -s start -c
```

where:

config_name

Is the name of the iSM configuration.

-d: Limits tracing to debug only.

For example:

```
iwsrv config_name -s start -d
```

where:

config_name

Is the name of the iSM configuration.

-f: [PATH] -- Filters the system path when invoking Java.

-f [RESTART] – Suppresses the JVM fault restart capability.

-h: Sets the iway7 home directory.

For example:

```
iwsrv config_name -s start -h C:\Program Files\iway7
```

where:

config_name

Is the name of the iSM configuration.

-t: The amount of time (in seconds) to process an iSM Windows service shutdown.

For example:

```
iwsrv config_name -s start -t
```

where:

config_name

Is the name of the iSM configuration.

Note:

- When running with the –l option, iSM currently relies on Java being in the path.
- The -l option should not be used in conjunction with iway7.cmd.
- The *iwsrv* command does not reliably stop the iSM Windows service.

Starting and Stopping iSM in a Command Window

For debugging purposes, you can run iSM in a command window instead of running it as a Windows service.

The command window enables you to interact with iSM and displays diagnostic information about iSM processing.

Procedure: How to Start iSM in a Command Window

To start iSM in a command window:

1. Open a Command Prompt window and navigate to the iSM home \bin directory. For example:

   ```
   C:\Program Files\iway7\bin
   ```

2. At the command prompt, type:

   ```
   iwsrv
   ```

   By default, the iSM *base* configuration is started.

   Note: To start other iSM configurations, type the iSM configuration name after the *iwsrv* command. For example:

   ```
   iwsrv TestConfig
   ```

   iSM configuration names are case-sensitive.

3. Press *Enter*.

   Messages are displayed in the command window detailing the iSM startup process. The following prompt is displayed when the iSM startup process is completed:

   ```
   Enter command:>
   ```

   If you receive the following error message, then iSM is probably already running as a Windows service:

   ```
   ERROR (HTTP1) Unable to create server socket 9980:
   java.net.BindException: Address already in use: JVM_Bind
   ERROR (SOAP1) Unable to create server socket 9000:
   java.net.BindException: Address already in use: JVM_Bind
   ```

   You cannot run the same iSM configuration at the same time as an iSM Windows service and in the command window.

Procedure: How to Stop iSM in a Command Window

To stop iSM in a command window:

1. Go to the iSM command prompt:

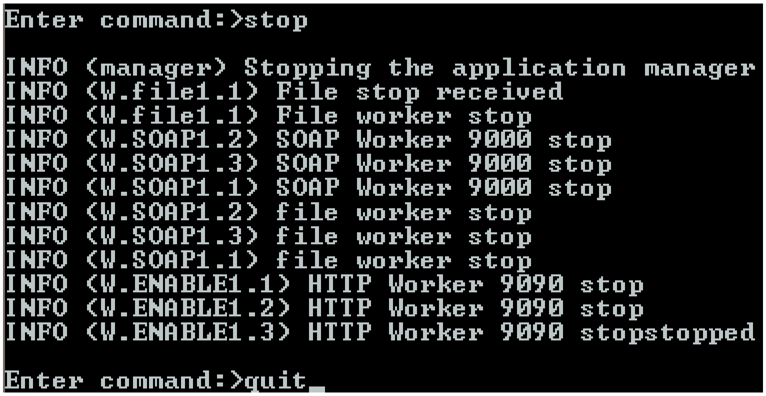
   ```
   Enter command:>
   ```

2. Type the following:

   ```
   stop
   ```

3. When any listeners that are running have stopped, type the following:

   ```
   quit
   ```



```
Enter command:>stop

INFO (manager) Stopping the application manager
INFO (W.file1.1) File stop received
INFO (W.file1.1) File worker stop
INFO (W.SOAP1.2) SOAP Worker 9000 stop
INFO (W.SOAP1.3) SOAP Worker 9000 stop
INFO (W.SOAP1.1) SOAP Worker 9000 stop
INFO (W.SOAP1.2) file worker stop
INFO (W.SOAP1.3) file worker stop
INFO (W.SOAP1.1) file worker stop
INFO (W.ENABLE1.1) HTTP Worker 9090 stop
INFO (W.ENABLE1.2) HTTP Worker 9090 stop
INFO (W.ENABLE1.3) HTTP Worker 9090 stopstopped

Enter command:>quit_
```

Note: Using the *quit* command terminates iSM. Never terminate iSM without using quit (for example, the *kill* command on Unix). Doing so may leave messages in process, resulting in a loss of resource integrity.

Installing and Deleting an iSM Windows Service

This section describes how to install and delete an iSM Windows service that can be used to start an iSM configuration.

By default, the *iwsrv* executable program installs an iSM Windows service to run out of process. You can view the options that were used to install an iSM Windows service by using the *iwsrv* query option. For more information, see *How to Query the Install Options for an iSM Windows Service* on page 410.

Procedure: How to Install an iSM Windows Service

To install an iSM Windows service:

1. Open a Command Prompt window and navigate to the iSM home \bin directory. For example:

   ```
   C:\Program Files\iway7\bin
   ```

2. At the command prompt, type

   ```
   iwsrv config_name -s install
   ```

   where:

   config_name

   Is the name of the iSM configuration for which you are creating a Windows service.

   To create a Windows service with Java running in a separate process, type

   ```
   iwsrv config_name -s install -l java
   ```

   where:

   config_name

   Is the name of the iSM configuration for which you are creating a Windows service.

   A message is displayed, indicating that the iSM Windows service was installed successfully.

   You must now start the iSM configuration as a Windows service. For more information, see *How to Start an iSM Configuration as a Windows Service* on page 408.

Procedure: How to Delete an iSM Windows Service

To delete an iSM Windows service that you created to start an iSM configuration:

1. Open a Command Prompt window and navigate to the iSM home \bin directory. For example:

   ```
   C:\Program Files\iway7\bin
   ```

2. At the command prompt, type

   ```
   iwsrv config_name -s remove
   ```

   where:

   config_name

   Is the name of the iSM configuration for which you are deleting a Windows service.

A message is displayed, indicating that the iSM Windows service was deleted successfully.

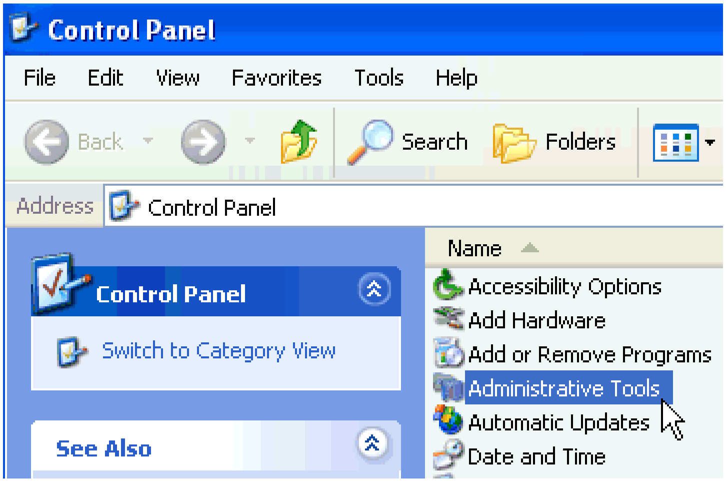
You can now delete the iSM configuration from the iWay Service Manager Administration Console. For more information, see *How to Remove a Configuration From the Console* on page 101.

Procedure: How to Start an iSM Configuration as a Windows Service

To start an iSM configuration as a Windows service:

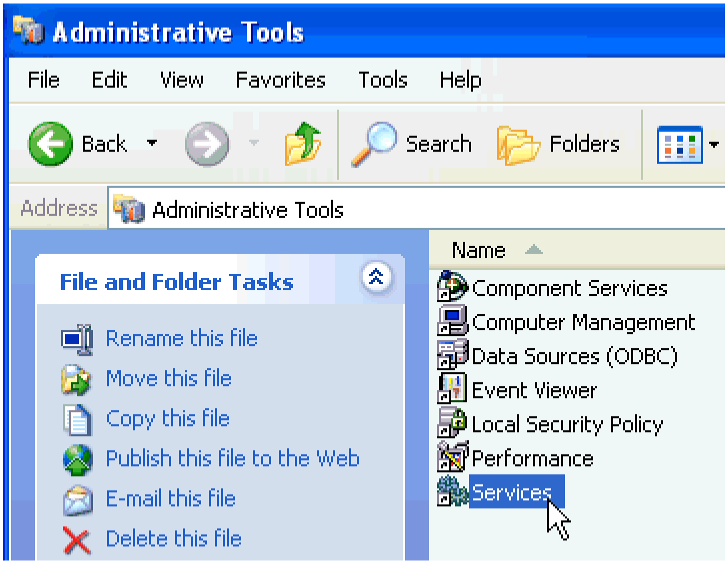
1. From the Windows Start menu, select *Control Panel*.

   The Control Panel opens.



2. Double-click *Administrative Tools*.

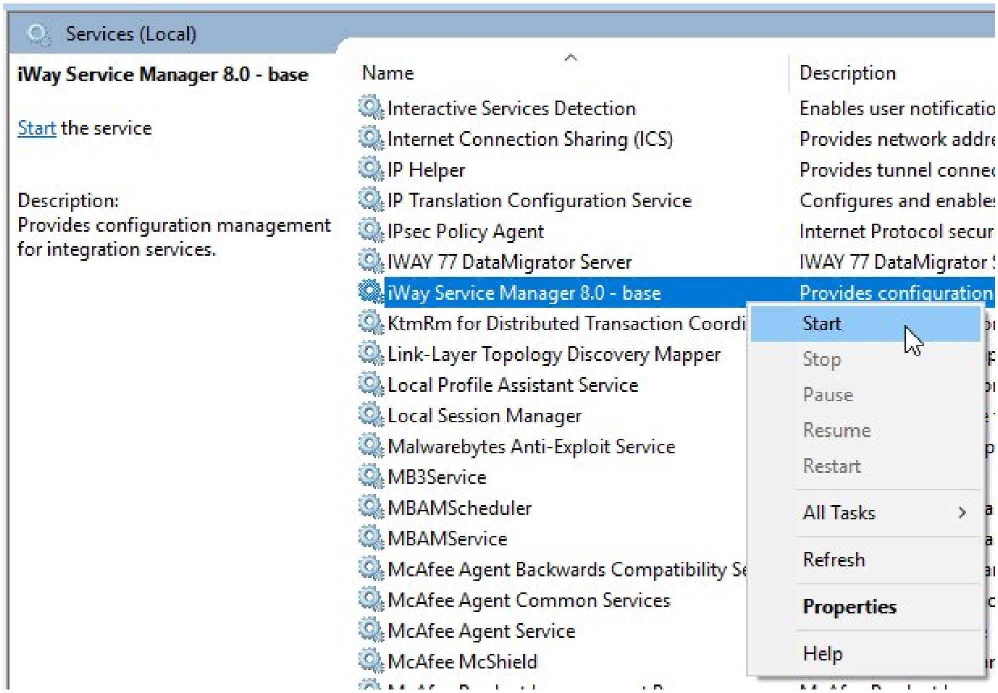
The Administrative Tools are displayed.



3. Double-click *Services*.

   The Windows Services are displayed.

4. Scroll down to display the iSM Windows service, as shown in the following image.



Services (Local)

iWay Service Manager 8.0 - base

Start the service

Description:
Provides configuration management for integration services.

Name	Description
Interactive Services Detection	Enables user notificatio
Internet Connection Sharing (ICS)	Provides network addr
IP Helper	Provides tunnel conne
IP Translation Configuration Service	Configures and enable
IPsec Policy Agent	Internet Protocol secur
IWAY 77 DataMigrator Server	IWAY 77 DataMigrator
iWay Service Manager 8.0 - base	Provides configuration
KtmRm for Distributed Transaction Coordi	
Link-Layer Topology Discovery Mapper	
Local Profile Assistant Service	
Local Session Manager	
Malwarebytes Anti-Exploit Service	
MB3Service	
MBAMScheduler	
MBAMService	
McAfee Agent Backwards Compatibility Se	
McAfee Agent Common Services	
McAfee Agent Service	
McAfee McShield	

5. Right-click the service you created (for example, *iWay Service Manager 6-1 - TestConfig*) and select *Start* from the context menu.

   The service Status changes to *Started*. The default Startup Type is *Automatic*.

Querying the Install Options for an iSM Windows Service

For debugging purposes, you can perform a query to display the options that were used to install an iSM Windows service.

Procedure: How to Query the Install Options for an iSM Windows Service

To query the install options:

1. Open a Command Prompt window and navigate to the iSM home \bin directory. For example:

   ```
   C:\Program Files\iway7\bin
   ```

2. At the command prompt, type

   ```
   iwsrv config_name -s query
   ```

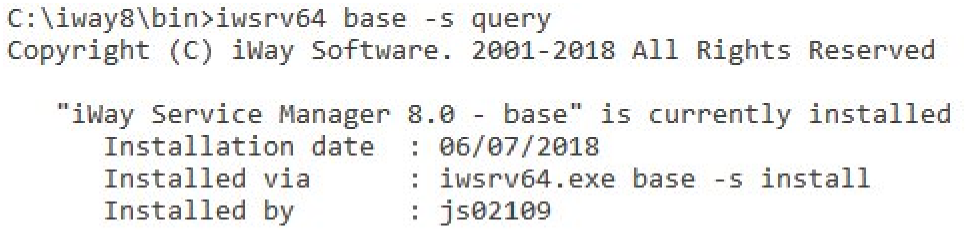
where:

config_name

Is the name of the iSM configuration for which you are running a query.

3. Press *Enter*.

   Information showing the options that were used to install the iSM Windows service is displayed in the command window. For example:



```
C:\iway8\bin>iwsrv64 base -s query
Copyright (C) iWay Software. 2001-2018 All Rights Reserved

   "iWay Service Manager 8.0 - base" is currently installed
      Installation date  : 06/07/2018
      Installed via      : iwsrv64.exe base -s install
      Installed by       : js02109
```

Understanding How IWSRV Selects a JVM in Process (Not -l Java)

This section describes how the *iwsrv* executable program selects a JVM in process (not -l Java).

1. If the IWAYISM7 environmental variable is set, then it will use its value to load the JVM (for example, G:\j2sdk1.4.1_03\jre\bin\server\jvm.dll).
2. If `-server` is specified as a JVM option, then the service uses the registry to find JAVAHOME of the Java Development Kit (JDK). Once JAVAHOME is determined (for example, G:\j2sdk1.4.1_03), then it appends the following string:

   ```
   \jre\bin\server\jvm.dll
   ```

3. The service uses the registry to determine where the current version of the JRE is installed. It queries for the following registry key:

   ```
   JavaSoft\\Java Runtime Environment\1.4\RuntimeLib
   ```

   The `1.4` component of the key is dynamic based on the current version of the JRE. The value of this key is then used to load the JVM. For example:

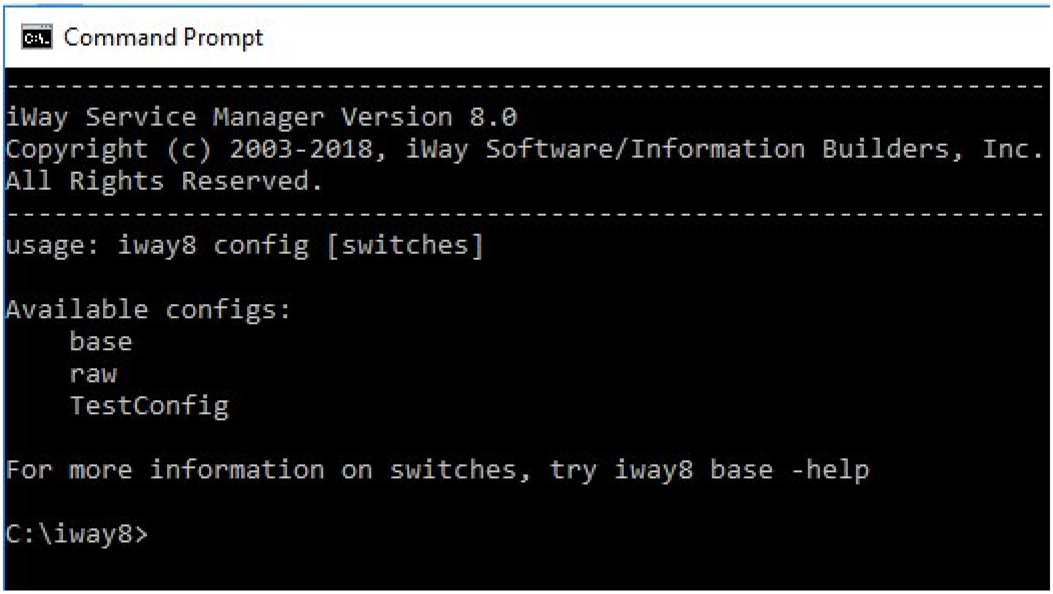
   ```
   E:\Program Files\Java\j2re1.4.1_03\bin\client\jvm.dll
   ```

4. During installation, you have the option to point to a JVM. If you choose this option, then the service uses this value to load the JVM.

Server Command Files

This section describes the available server command files that you can use to manually control startup activity.

You can use iWay Service Manager (iSM) to execute commands, such as *start* and *stop*, from the iSM command shell, as shown in the following image.



```
-----------------------------------------------------------------
iWay Service Manager Version 8.0
Copyright (c) 2003-2018, iWay Software/Information Builders, Inc.
All Rights Reserved.
-----------------------------------------------------------------
usage: iway8 config [switches]

Available configs:
    base
    raw
    TestConfig

For more information on switches, try iway8 base -help

C:\iway8>
```

These commands can also be added to command files and executed by the following:

- ❏ A `run <filename>` command from the shell or from another command file.
- ❏ The server at startup, by using the autocmd.txt command file in the startup directory.

All server commands that are available from the shell can be included in command files. iWay expects that those commands that affect the server, such as *start*, *stop* and *set* will be the most useful. Nothing prevents other commands, such as *memory*, from being run in the command file. However, the information that is output by these commands may not appear in a convenient place.

Note: To control which JRE is used by the iWay Secure Messaging Option (iSMO) product, you can set the IWAYISM7 system environment variable to the full path where the jvm.dll file is located. For example:

```
IWAYISM7=c:\java\jre\bin\server\jvm.dll
```

Understanding the Command File Structure

Command files can make use of the iWay Functional Language during their execution. For example, consider a properties file names *testprop* that contains the following lines:

```
dostart=start
whattoday=chan3
```

Now consider the following command file:

```
1. // this is a test file
2. set debug on
3. _property('c:/testprop.properties','dostart',';') file1
4. _if(sreg('condition1')='true'),'start file1',';')
5. // last line of test file
```

The first line is a comment. Any line beginning with a // or a semicolon is a comment. The second line enables debug tracing. Line 3 is an IFL statement, testing the *dostart* property in the *testprop.properties* file. If the property is not found, then the semicolon is returned and the line is ignored. If it is found, the value *start* is returned, and the line now reads:

```
start file1
```

This line starts the named channel *file1*.

Line 4 is another IFL statement. In this case, if the special register condition1 is set to true, then the start file1 command is executed.

Line 3 and line 4 show two methods for conditionally executing lines in a command file.

The command file also supports the special command goto <label>. A label is a named line in the file located after the *goto* command. A label is defined as a Java defined name that ends with a colon and must exist with no other commands on that line.

```
1. // this is a test file with goto
2. goto point
3. _if (_properties('c:/props','whattoday')=='chan3','goto point3')
4. start chan1
5. start chan2
6. goto end
7. point3:
8. start chan3
9. end:
10. // last line of test file
```

In this command file, the IFL statement on line 3 instructs the server that if the *whattoday* property is set to chan3, then the commands pick up on line 7. Else, the commands on lines 4 through 6 will execute. For more information on using a command file, see the *iWay Service Manager Command Reference Guide*.

Configuring the Autocmd.txt File

The *autocmd.txt* file can be configured to run during the iSM startup process to set customized options for use by iSM. For example, this configuration file can be used to map network drives on your system.

To enable this functionality, the *autocmd.txt* file must be copied to an iSM configuration directory. For example:

```
C:\Program Files\iway7\config\base
```

The contents of the *autocmd.txt* file can point to a batch file on your file system. For example:

```
shell C:\a1.bat
```

The batch file (for example, a1.bat) can contain the following network drive assignments:

```
net use W: /delete > c:\iway7\netuse.txt 2>&1
net use W: \\ibiprda\IBI  xxxx /USER:ibi\XX12345 /persistent:yes >> c:
\iway7\netuse.txt 2>&1
```

Channel Startup Order

iWay Service Manager (iSM) startup always first creates internal (and ordered) queues, so that they are available as soon as channels that may need to be added to the queue begin processing messages. Next, active channels are started. If the channel is configured with a startup dependency, then the dependent channels are started before the channel itself. This way, you can control the startup sequence.

A startup script (or flow) can also start channels in any order that is required.

Event Process Flows

Event process flows can be executed when specific (defined) events occur in iWay Service Manager (iSM) or during message processing. The process flows must be published to the configuration (iWay Integration Application) and must be available for execution at the time that they are called.

The Event process flows can run under the following constraints:

❏ Communicate with the caller by passing a return code as the name of the End node. This is the same rule as is required for subflows of a regular process.

❏ Can only return a single document, which may or may not be meaningful to the caller.

❏ Cannot use Emit nodes, although Emit services are permitted. Emit nodes schedule emits for execution at a later time (asynchronous to the process flow), while Emit services emit directly when they are called.

Other restrictions may apply for individual Event process flows. All Event process flows are conditional, and must be configured for execution if their use is required.

The following Event process flows are described in this section:

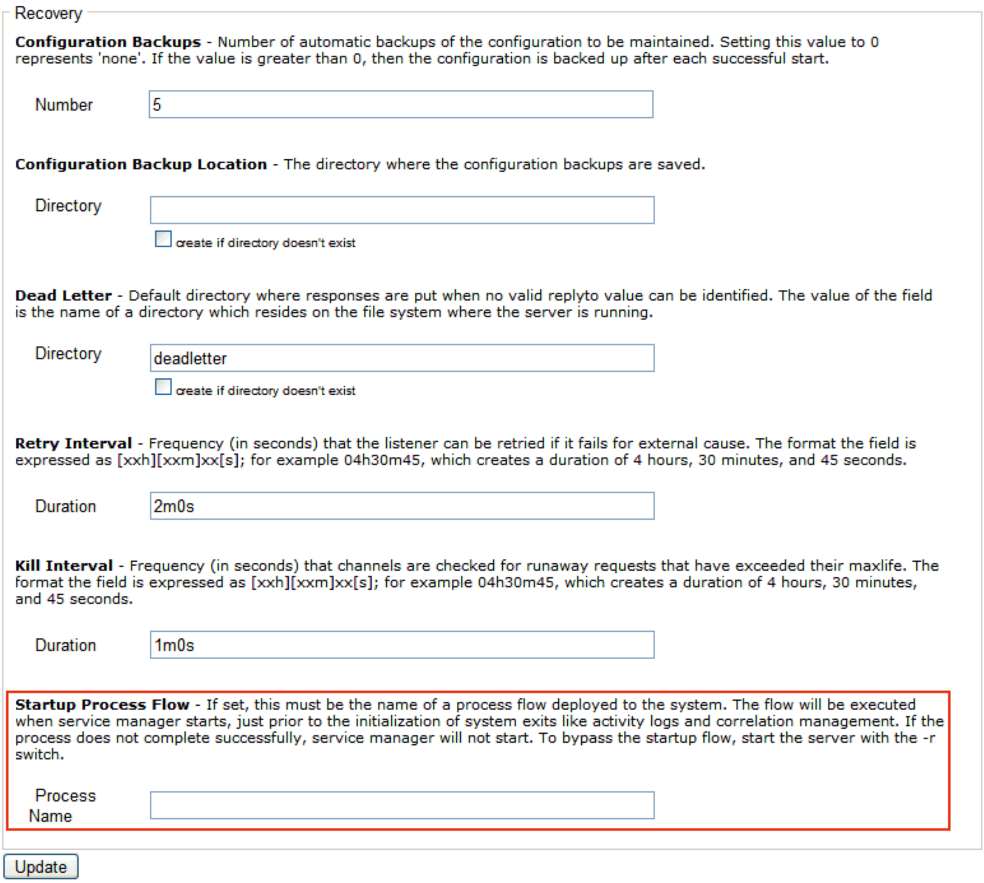
- Server Startup
- iWay Business Activity Monitor (BAM) Database Loss of Access
- Channel Startup Failure
- Retry Expired
- Failed ReplyTo
- Send to Dead Letter
- Parse Failure

Server Startup

The Server Startup process flow is executed by the iSM initialization routines as iSM starts its execution. This process flow can check for the availability of resources that are required by iSM, and can prevent iSM from starting if the resources are not available. A return of *success* allows iSM to continue its startup sequence. Otherwise the iSM startup is terminated.

The Server Startup process flow cannot start channels, since iSM is not ready to run channels at this early (startup) stage.

The name of the Server Startup process flow must be entered in the Recovery area of the General Settings page (Process Name field), as shown in the following image.



Recovery

Configuration Backups - Number of automatic backups of the configuration to be maintained. Setting this value to 0 represents 'none'. If the value is greater than 0, then the configuration is backed up after each successful start.

Number: 5

Configuration Backup Location - The directory where the configuration backups are saved.

Directory:

□ create if directory doesn't exist

Dead Letter - Default directory where responses are put when no valid replyto value can be identified. The value of the field is the name of a directory which resides on the file system where the server is running.

Directory: deadletter

□ create if directory doesn't exist

Retry Interval - Frequency (in seconds) that the listener can be retried if it fails for external cause. The format the field is expressed as [xxh][xxm]xx[s]; for example 04h30m45, which creates a duration of 4 hours, 30 minutes, and 45 seconds.

Duration: 2m0s

Kill Interval - Frequency (in seconds) that channels are checked for runaway requests that have exceeded their maxlife. The format the field is expressed as [xxh][xxm]xx[s]; for example 04h30m45, which creates a duration of 4 hours, 30 minutes, and 45 seconds.

Duration: 1m0s

Startup Process Flow - If set, this must be the name of a process flow deployed to the system. The flow will be executed when service manager starts, just prior to the initialization of system exits like activity logs and correlation management. If the process does not complete successfully, service manager will not start. To bypass the startup flow, start the server with the -r switch.

Process Name:

The following table lists and describes the possible edges that are returned by the Server Startup process flow.

Edge	Description
success	Continue with iSM startup.
<other> or flow fails	Do not continue to start iSM.

iWay Business Activity Monitor Database Loss of Access

This process flow is executed when the iWay Business Activity Monitor (BAM) drivers lose connectivity to the BAM database. The process flow can notify an operation area of the problem, and can determine how iSM should continue:

- ❏ iSM continues, but BAM update is ignored.
- ❏ iSM terminates.
- ❏ iSM maintains a local file on disk containing BAM information and attempts to update the database when connectivity is restored.

Channel Failure

The channel failure process flow is executed if the channel fails during its normal run cycle. iSM monitors the channels, and attempts to intercept failures within the channel that are not handled by normal channel and server error management. If such an error is detected, the *ChannelFailure* exit is entered. The input document to the process flow signals the cause of the failure, along with statistical and other information as available. This process flow must be published to the system, and is executed whenever the channel encounters an error that cannot be handled by normal logic. The process flow is often used to send an email to alert an administrator of the issue.

iSM divides channel operation into three parts:

- ❏ Initialization
- ❏ Startup
- ❏ Message Handling

The channel failure process flow applies for initialization and startup only to channels that are not started by a specific manual command. This is because the operator is usually starting the channel and is monitoring its startup.

The message handling section of the channel is itself divided into the following three parts:

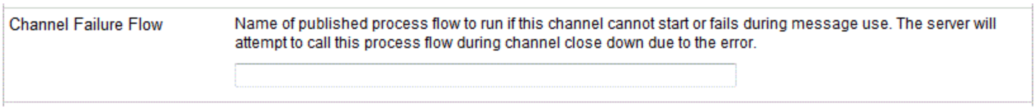
- ❏ **Message Acquisition.** A message is obtained from an external source, such as an MQ queue, an internal queue, a file, HTTP, and so on. The message is prepared for execution.
- ❏ **Message Processing.** The actual handling of the message itself. This includes the associated process flow as well as other steps such as preparsing, reviewing, emitting responses, reencoding, and so on. The process flow is itself transactional around its actions. iWay refers to the process flow transaction as the *inner transaction*, because it is taking place during the internal message handling.

- **Message Disposition.** The message that has been handled is committed. This can include sending a response code to an HTTP source, committing the read of the message from a queue (removing it), or deleting the file that has been read and processed. This is referred to by iWay as the *outer transaction*.

iWay treats the inner and outer transactions independently, as a failure of the inner transaction is application dependent while the outer transaction depends on the message environment. If the two transactions combined, a message cane become lost. However, this is prevented by the transaction model. For some protocols, failure of the outer transaction can trigger a special event flow called the *Outer Transaction Failure Flow*. The channel itself has not failed, but the specific message may require special handling.

This flow is invoked when a message has been processed by the channel, but the listener is not able to complete processing by eliminating the original message. For example, when an MQ listener is not able to commit the read from the queue.

To configure a channel failure process, enter the name of the published process flow to run in the Channel Failure Flow field of your listener configuration, as shown in the following image.



Channel Failure Flow

Name of published process flow to run if this channel cannot start or fails during message use. The server will attempt to call this process flow during channel close down due to the error.

Note: Although iWay attempts to maintain document structure, the nature of this channel failure process flow opportunity may require changes at any time. iWay will attempt to guarantee the structure between releases and service packs, but additional information may be added at any time. Any such changes will be documented in the release or patch release notes.

The channel failure process flow receives a signal message document for processing. The signal message document uses the following structure and format:

```
<channelfail name="File" state="4" statename="active"
time="2013-09-20T16:18:11Z" type="unexpected" version="7.0.0">
  <exception>
    <cause>java.lang.NullPointerException</cause>
    <stacktrace>
      java.lang.NullPointerException&#xd;
      at com.ibi.edaqm.XDFileMaster.run(XDFileMaster.java:301)&#xd;
      at java.lang.Thread.run(Thread.java:724)&#xd;
    </stacktrace>
  </exception>
  <statistics completed="177" failed="2"/>
</channelfail>
```

A startup failure process flow may receive the following document:

```
<channelfail name='channelname' protocol='protocol'
state='statecode'statename='init'  failures='2' version='7.0.0'=
time='2013-09-20T16:18:11Z' type='init'>
    <message>Cannot locate file directory</message>
</channelfail>
```

The *type* attribute identifies the form of the failure trap, which is listed and described in the following table.

Attribute	Applies to Manual Start?	Description
unexpected	Yes	An unhandled runtime exception was encountered at some stage of channel execution. Usually this indicates a programming problem and should be reported to iWay Customer Service.
retry	No	An error was encountered, but it was handled directly by the channel. The channel elected to set itself into a *retry* state, meaning that it will be restarted by iSM following a configured wait time.
init	No	The channel could not be initialized. Usually this means the resources needed could not be located.
start	No	The channel could not be started. Usually this phase attempts to actually access resources.

The process flow can return instructions to iSM to control special handling. This is accomplished by ending the process flow on an End node of one of the names that is listed and described in the following table.

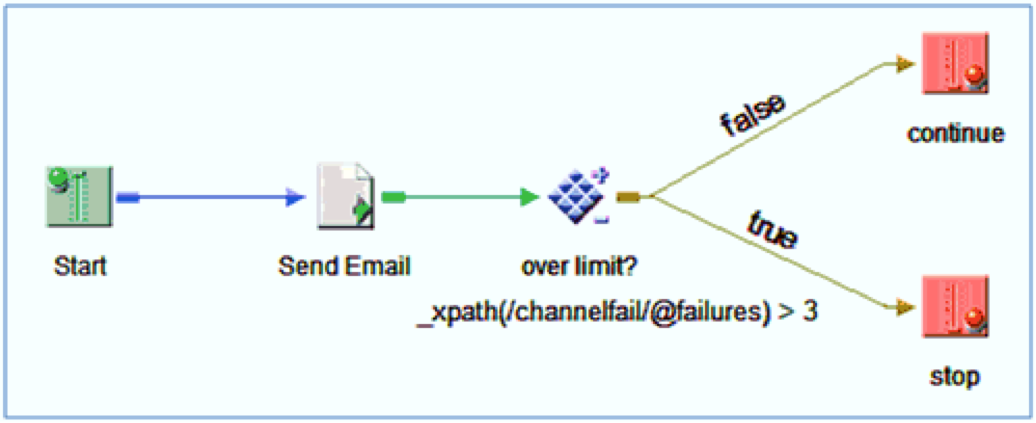
End Node Name	Applies To Attribute	Description
config	unexpected, retry, start, and init	Leave the channel in a configuration error state. Retry will not be attempted, and a manual start is not accepted.

End Node Name	Applies To Attribute	Description
stop	unexpected, retry, start, and init	Place the channel in a stopped state. A start command is required to restart the channel.
retry	unexpected and retry	Place the channel into a retry state. This is the default action for a retry or startup event.

Information in the *channelfail* document is listed and described in the following table.

Parameter	Description
name	The name of the configured channel to which this flow applies.
statename	The current state of the channel, which can be set by the process flow. ❑ **config.** Cannot start due to a configuration error. The channel start will not be retried. ❑ **retry.** The channel will attempt to restart after a configured retry interval. ❑ **stopped.** The channel is not running and can only be restarted by a specific (manual or programmed message) command.
state	The internal code that refers to the channel state. The statename usually explains this, and the code is provided to assist in issue isolation.
protocol	The protocol of the channel (for example, File and MQ.)
failures	Applies only to *init* and *start* entry. This is the number of sequential failures encountered during the channel startup cycle.
version	The version of the ism server, and thus this document.
time	The time that the document was created.
statistics	For runtime retry, this is the number of messages processed.

In the following simplified example, a failure results in an email being sent to an identified party followed by a check to see if the number of sequential failures exceeds a designated limit (in this example, 3).



Normally this process flow would run during iSM startup or channel restart. To have the process flow run if the start is attempted from an iSM *start* command whether standalone or in a script, use the *-doflow* switch on the start command.

Outer Transaction Failure Flow

This event flow is offered by certain channels, such as MQ. It is invoked when the acquired message has ben handled by the appropriate process, but the elimination of the message itself (for example, committing the read from the queue) fails. If this occurs, the acquired message can be reacquired by the iSM channel as a possible duplicate.

The message received by the event flow is similar to the message received by the channel failure flow, except that the input document is of type *outercommitfail.*

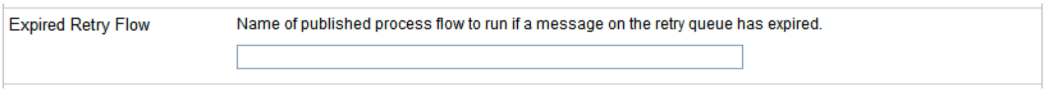
The event flow can return *stop* to signal the server to attempt to terminate the channel upon completion of any messages currently in the channel. Stopping triggered by the event flow does not guarantee that no more messages will be processed, since other workers might be busy with their own messages when the channel receives the stop signal.

Retry Expired

Messages can be queued for retry on channels that support this facility. This includes queue-based channels, the File channel, and the Internal Queue channel. The retries are triggered by logic in the process flow. In this circumstance, the message is re-executed on a periodic basis until expiration has been reached.

At the expiration point, a process flow can be executed to take recovery actions including notification, and optionally, changing the destination address or restarting with a changed (extended) expiration time.

Enter the name of a published process flow to be executed in the Expired Retry Flow field, as shown in the following image. This field is a common channel property that is available for all iSM listeners.



Expired Retry Flow

Name of published process flow to run if a message on the retry queue has expired.

On entry, the process flow receives the document as it exists, at the point at which the process flow is called. The following table lists and describes several special registers that are available in the Retry Expired process flow to assist during the analysis.

Register Name	Description
iway.eventflow.exitflow	Identifies the purpose of the process flow (for example, *expiredRetry*).
iway.eventflow.attempts	Count of the number of retry attempts made before the expiration.
iway.eventflow.expiredtime	Time of the expiration.

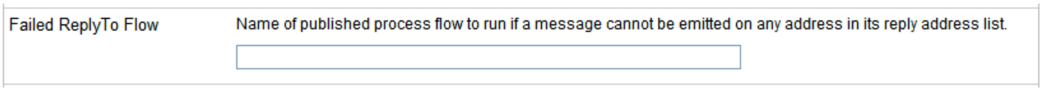
The following table lists and describes the possible edges that are returned by the Retry Expired process flow.

Edge	Description
success	The process flow overrules the expiration. iSM will attempt to resend, this time with the output of the process flow.
<other> or flow fails	An error document is sent to the error addresses.

Failed ReplyTo

A reply designation associated with a document triggers an emit operation following completion of the process flow. If the emit operation is not successful, the Failed ReplyTo process flow is triggered.

Enter the name of a published process flow to be executed in the Failed ReplyTo Flow field, as shown in the following image. This field is a common channel property that is available for all iSM listeners.


Failed ReplyTo Flow

Name of published process flow to run if a message cannot be emitted on any address in its reply address list.

On entry, the process flow receives the document as it exists, at the point at which the process flow is called. The following table lists and describes several special registers that are available in the Failed ReplyTo process flow to assist during the analysis.

Register Name	Description
iway.eventflow.exitflow	Identifies the purpose of the process flow (for example, *failedReply*).
iway.eventflow.replyname	Configured name of the reply or error specification.
iway.eventflow.destination	The address configured for the emit, as evaluated for use.
iway.eventflow.errormsg	An error message, if any, describing the cause of the failure that caused this event to be generated.
iway.eventflow.replyprotocol	Protocol used for the emit attempt (for example, File, MQ, and so on).

The following table lists and describes the possible edges that are returned by the Failed ReplyTo process flow.

Edge	Description
success	The process flow took responsibility to deliver the message.
<other> or flow fails	An error document is sent to the error addresses.

Each ReplyTo and ErrorTo is treated separately. If an error occurs for one, an attempt is made to handle the error, and iSM continues with the rest of the list. Error handling, however, differs for ReplyTo versus ErrorTo.

A failed ReplyTo causes the Failed ReplyTo process flow to execute (if present). If the process flow is successful (by terminating at an End node called success), the error is considered to be handled and iSM continues through the rest of the address list. If the process flow is absent, fails, or reaches an End node with a different name, then iSM creates an error document and attempts to send it to the ErrorTo instances recursively. All ErrorTo instances will be called for each ReplyTo that fails.

Document siblings are treated as independent documents. The net effect should be similar to sending the document first, and then each of its siblings one by one. iSM does not expect error documents to contain siblings. However, if present, they too will be sent as top-level documents (which may or may not be in error).

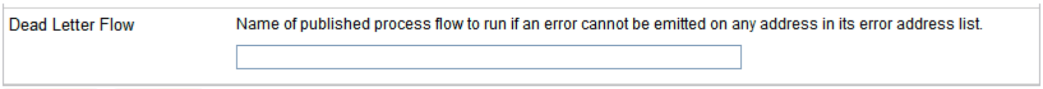
Send to Dead Letter

Messages queued for emitting at a later time, using the ReplyTo and ErrorTo channel configurations, or the Emit object in a process flow, are sent when the outlet of the channel is executed. Messages can also have alternate addresses if required.

If all attempts to emit the message fail, then by default, the message is written to a configured *dead letter* directory.

If an *emit failed* process flow is configured, then the process flow can examine the message, redirect it, replace it, and potentially notify an appropriate authority. It can then send the message to another channel for a retry attempt or continue to allow the message to be written to the dead letter queue.

Enter the name of a published process flow to be executed in the Dead Letter Flow field, as shown in the following image. This field is a common channel property that is available for all iSM listeners.



Dead Letter Flow

Name of published process flow to run if an error cannot be emitted on any address in its error address list.

On entry, the Send to Dead Letter process flow receives the document as it exists at the point at which the process flow is called. The following table lists and describes several Special Registers (SREGs) that are available in the process flow to assist during the analysis.

Register Name	Description
iway.eventflow.exitflow	Identifies the purpose of the process flow (for example, *deadLetter*).

Register Name	Description
iway.eventflow.replyname	Configured name of the reply or error specification.
iway.eventflow.destination	The address configured for the emit, as evaluated for use.
iway.eventflow.errormsg	An error message, if any, describing the cause of the failure that caused this event to be raised.
iway.eventflow.replyprotocol	Protocol used for the emit attempt (for example, File, MQ, and so on).

The following table lists and describes the possible edges that are returned by the Send to Dead Letter process flow.

Edge	Description
success	The message was successfully handled.
<other> or flow fails	The output of the process flow to be written to the dead letter directory, if configured.

Each ReplyTo and ErrorTo is treated separately. If an error occurs for one, an attempt is made to handle the error, and iSM continues with the rest of the list. Error handling, however, differs for ReplyTo versus ErrorTo.

A failed ReplyTo causes the Failed ReplyTo process flow to execute (if present). If the process flow is successful (by terminating at an End node called success), the error is considered to be handled and iSM continues through the rest of the address list. If the process flow is absent, fails, or reaches an End node with a different name, then iSM creates an error document and attempts to send it to the ErrorTo instances recursively. All ErrorTo instances will be called for each ReplyTo that fails. ErrorTo instances are used to communicate errors to administrators who are able to resolve such situations.

A failed ErrorTo causes the Send to Dead Letter process flow to execute (if present). If the process flow returns success, iSM considers the error to be handled and continues with the rest of the address list. If the process flow is absent, fails, or reaches an End node with a different name, then iSM attempts to write a file under the configured dead letter directory.

Sending an error to an empty list of ErrorTo instances is an error. It is handled the same way as a failed ErrorTo.

Notice that only error documents are sent to the configured dead letter directory. If an error cannot be reported (because an ErrorTo fails or there are no ErrorTo instances), then iSM attempts to send the error document to the dead letter directory to keep a record for manual processing. An error document contains a copy of the original document that generated the error.

iSM attempts to avoid sending to a duplicate address within the list if iSM already knows it is a bad address. This could happen when an ErrorTo is also a ReplyTo. A duplicate bad address is treated the same as a regular failed ReplyTo or ErrorTo, except the IO was never attempted.

Document siblings are treated as independent documents. The net effect should be similar to sending the document first, and then each of its siblings one by one. iSM does not expect error documents to contain siblings. However, if present, they too will be sent as top-level documents (which may or may not be in error).

Parse Failure

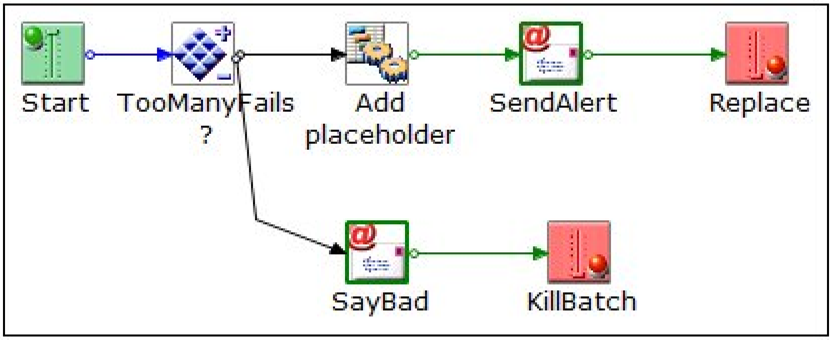
The Parse Failure flow is invoked if an incoming message fails the *parse to XML* operation for a channel. This does not apply to a parse that is handled within a process flow by a service (agent) for that purpose.

The incoming document to the flow contains the message that failed parsing. The standard Special Registers (SREGs) for the protocol are available in the flow. For example, a bad message on a File listener will provide the usual information on the source of the file.

The Parse Failure flow can also be used to send a notification.

The flow can replace the document that could not be parsed. This might be done to *fill in* an element in a large batch managed by a splitting preparser. To replace the message, set the document on output to the message required, and return through an End node named *Replace*. The replaced message will then pass through the normal channel cycle. It may be necessary in your application to set a SREG in order to notify subsequent processes that this is a *placeholder* message. If this technique is used, then remember to set the SREG at the channel level, so as to make it available beyond the scope of the flow.

On entry to the event flow, the SREG *iway.parsefail* will be set to the count of the number of parse failures in this channel for this transaction. This count is useful for batch handling, in which a splitting preparser divides the batch into a sequence of sub-messages. For example, your flow might determine that the count of *placeholder* messages returned to the channel has exceeded a threshold, and so elects to take application action to reject the batch.



Startup Process Flow

The startup process flow optionally executes as iSM starts. The name of the process flow is entered in the *Recovery* area of the iSM Administration Console. Click *Server*, then *General Settings* from the Settings group on the left pane, and then scroll down to the *Recovery* area.

If named and deployed, the process flow is executed by the server just prior to the installation of system components. For example, if SNMP did not begin, then the process flow itself will not be recorded in the activity logs.

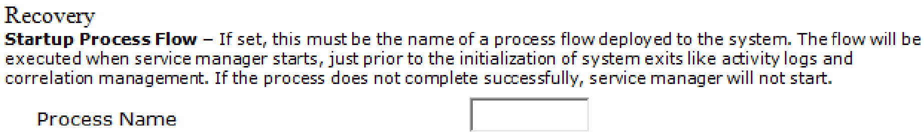
If the process flow ends successfully, the server continues with its startup process. If the process does not end successfully (for example, a fail service is encountered), the server does not start.

The process flow is designed to enable the server to verify the availability of required resources. For example, an SQL service in the process flow may perform a simple select against the Business Activity Monitor (BAM) tables by accessing the jdbc/BAMDBProvider. If the select fails, it can be assumed that the BAM database is not available, and the process flow issues a fail. This would prevent processing if BAM, deemed by the application designer to be a critical resource, is not available. Similarly, if an application required the transfer of data from an Oracle to a DB2 database, the startup process flow could determine that both are available before allowing the server to start. Startup criteria are at the discretion of the application designer.

Once started, the server manages errors and recovery normally.

You cannot control the server from this process flow. For example, you cannot use the control service to start channels because the server has not yet been sufficiently initialized for channels to properly start. Other facilities, including the autostart script, can be used for this purpose.

The following image shows the Recovery pane.



Recovery

Startup Process Flow – If set, this must be the name of a process flow deployed to the system. The flow will be executed when service manager starts, just prior to the initialization of system exits like activity logs and correlation management. If the process does not complete successfully, service manager will not start.

On entry, the input document to the process flow is shown below:

```
<startup version=currentversion time=timestamp/>
```

where:

currentversion

Is the server version number, such as 7.0

timestamp

Is a standard RFC 3339 (ISO 8601) timestamp.

The output document is ignored.

The startup parameter -r causes iSM to start without calling the startup exit. This allows a "buggy" startup exit to be bypassed so that iWay tools can be used to correct any problems.

Note: This startup parameter is available under the batch (manual) startup mode. Users are advised to avoid starting as a service until the startup exit is known to be functioning properly.

Closedown Process Flow

The closedown process flow, if configured, is executed immediately before iSM is shut down. All listeners and channels are closed, but providers remain available. The result of this process flow is ignored, and a shut down of iSM proceeds regardless of the actions taken by the process flow. In a sample scenario, the closedown process flow can be used to send a notification to an administrator that iSM has shut down, releasing resources as a result, and so on. You can use an internal emit to send a *start up* message to a persistent internal queue to be acted upon whenever iSM is started again. You cannot use the Controls Channel Service (com.ibi.agents.XDControlAgent) to manage channels or otherwise affect iSM.

SNMP Provider

SNMP (Simple Network Management Protocol) is a standard Internet protocol to monitor attached devices for conditions or situations that warrant attention. Within a managed network, iWay Service Manager (iSM) is treated as a device to be monitored.

Under SNMP, the two key components are a manager that aggregates, evaluates, and displays information from agents, which represent the managed devices. Typically an iSM user will already have a manager installed for network control. The manager is an external, non-iWay component to which iSM reports. Managers range from simple freeware versions that are downloadable from the web to sophisticated management systems from major vendors.

SNMP exposes management data from the agent in the form of described variables on the managed systems devices. These variables can then be queried (and sometimes set) by the manager.

The SNMP standard does not define which information (variables) a managed device should offer. SNMP uses an extensible design, where the available information is defined by management information bases (MIBs). MIBs describe the structure of the management data of a device subsystem.

When exposing information using SNMP, iSM assumes the role of a managed device. The SNMP provider acts as the SNMP agent. A provided MIB details the information that iSM exposes, including:

- server start and end
- listener start and end
- listener execution statistics including messages processed and their execution time
- special registers (SREGs)

You may have a requirement to monitor iSM along with Java and operating system values. The Java Virtual Machine (JVM) and most operating systems offer MIBs that can also be loaded into your manager. You can then combine these to provide a more complete report of server activity and resource use.

The SNMP remote function calls (RFCs) describe three protocol standards, which are known as V1, V2c, and V3. Managers can choose to implement one or more of these standards. The iSM SNMP provider supports all three standards.

The SNMP facilities are automatically installed during the iSM installation. The MIBs for iSM are located in the following directory:

```
<iwayhome>/etc/mibs
```

The MIBs are copied to the manager software (and usually compiled) as required by that software. The MIBS for Java and the operating system, can be found according to their own software installation. For example, the Java MIB provides access to Java execution information such as threads, memory use, and semaphores (monitors).

Configuring the SNMP Provider

The iSM SNMP agent is exposed as a provider. There can be one or more providers defined in a configuration, although more than one is only required if there are independent managers controlling different aspects of the server.

To complete the configuration, you will need to know the configuration of the manager. For example, the manager will be configured to interact with the agent on a specific port (usually 161). If you have multiple configurations on the same installation host, you will need to separate these by having different addresses. For performance reasons, iSM uses separate providers on each configuration rather than having a single provider poll other configurations.

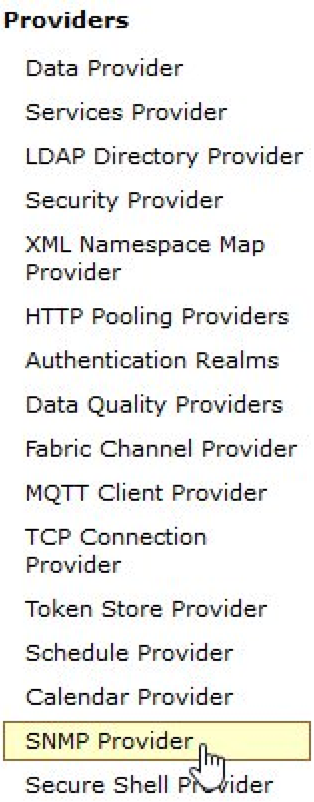
You will also be requested to specify which protocol(s) are used by your manager. Select the set that is supported by the manager.

The provider can accept simultaneous requests from multiple managers, and some managers are capable of sending multiple requests to their agents in parallel. If this is the case, you can specify the number of expected parallel requests by setting the number of execution threads.

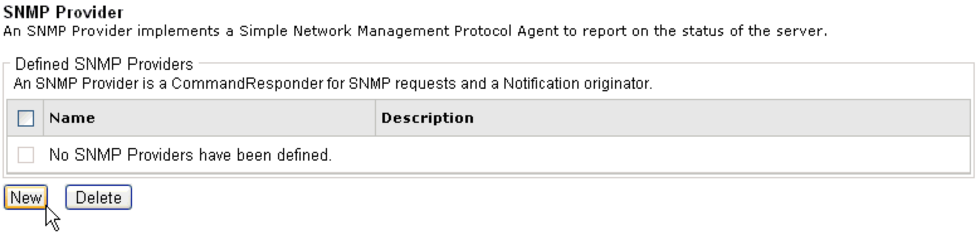
Procedure: How to Configure the SNMP Provider

To configure the SNMP provider:

1. In the left console pane of the Server menu, select *SNMP Provider*.



The SNMP Provider pane opens.



SNMP Provider

An SNMP Provider implements a Simple Network Management Protocol Agent to report on the status of the server.

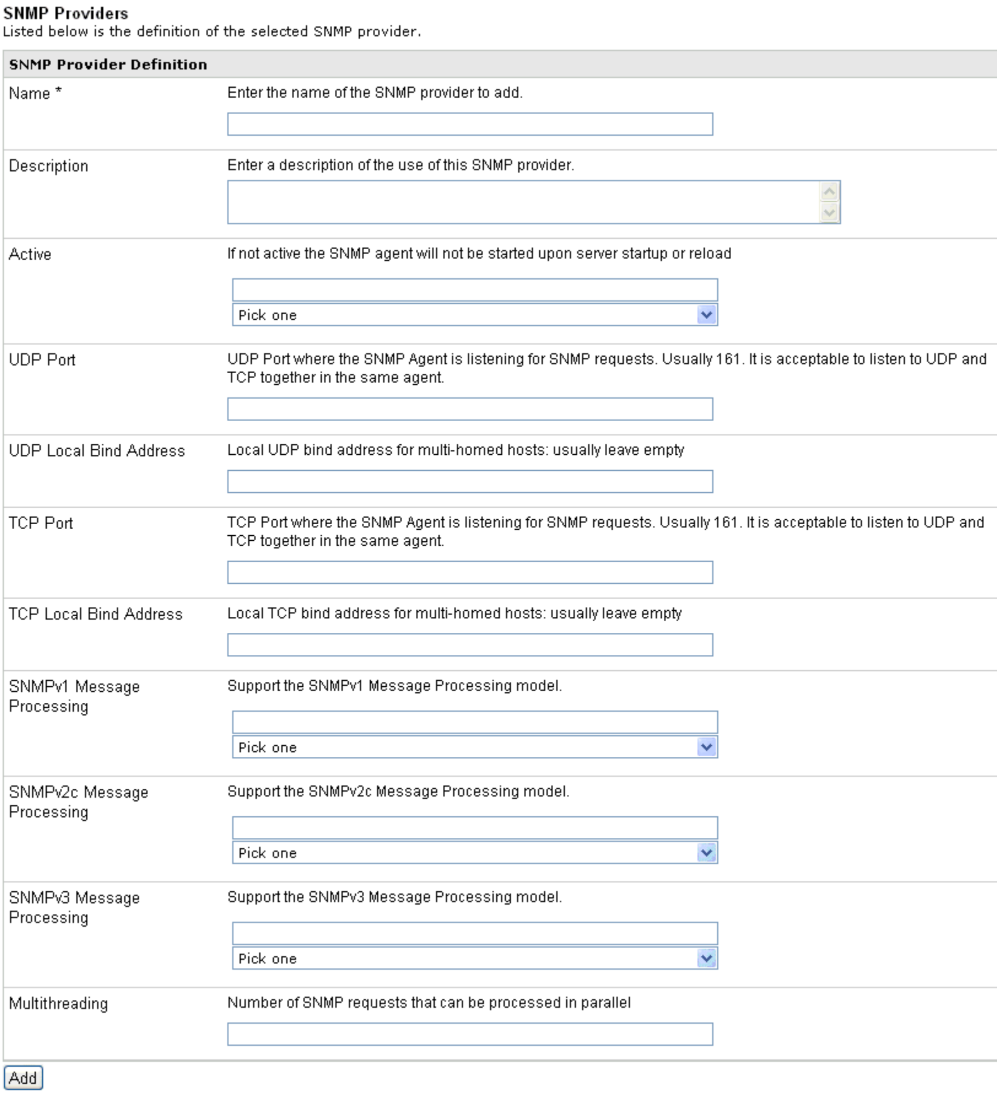
Defined SNMP Providers

An SNMP Provider is a CommandResponder for SNMP requests and a Notification originator.

□	Name	Description
□	No SNMP Providers have been defined.	

2. Click *New* in the Defined SNMP Providers section.

The SNMP Provider Definition pane opens.



SNMP Providers
Listed below is the definition of the selected SNMP provider.

SNMP Provider Definition	
Name *	Enter the name of the SNMP provider to add.
Description	Enter a description of the use of this SNMP provider.
Active	If not active the SNMP agent will not be started upon server startup or reload Pick one
UDP Port	UDP Port where the SNMP Agent is listening for SNMP requests. Usually 161. It is acceptable to listen to UDP and TCP together in the same agent.
UDP Local Bind Address	Local UDP bind address for multi-homed hosts: usually leave empty
TCP Port	TCP Port where the SNMP Agent is listening for SNMP requests. Usually 161. It is acceptable to listen to UDP and TCP together in the same agent.
TCP Local Bind Address	Local TCP bind address for multi-homed hosts: usually leave empty
SNMPv1 Message Processing	Support the SNMPv1 Message Processing model. Pick one
SNMPv2c Message Processing	Support the SNMPv2c Message Processing model. Pick one
SNMPv3 Message Processing	Support the SNMPv3 Message Processing model. Pick one
Multithreading	Number of SNMP requests that can be processed in parallel

3. Provide the appropriate values for your SNMP provider parameters as listed and defined in the following table.

Parameter	Description
Name *	Enter the name of the SNMP provider to add.

Parameter	Description
Description	Enter a brief description of the of the SNMP provider.
Active	If set to *false*, the SNMP agent will not be started during iSM server startup or restart.
UDP Port	The UDP port where the SNMP agent is listening for SNMP requests. Usually 161. It is acceptable to listen to UDP and TCP together in the same agent.
UDP Local Bind Address	The local UDP bind address for multi-homed hosts. This parameter value is usually left blank.
TCP Port	The TCP port where the SNMP agent is listening for SNMP requests. Usually 161. It is acceptable to listen to UDP and TCP together in the same agent.
TCP Local Bind Address	The local TCP bind address for multi-homed hosts. This parameter value is usually left blank.
SNMPv1 Message Processing	Select *true* to support the SNMPv1 message processing model.
SNMPv2c Message Processing	Select *true* to support the SNMPv2c message processing model.
SNMPv3 Message Processing	Select *true* to support the SNMPv3 message processing model.
Multithreading	The number of SNMP requests that can be processed in parallel.

Community

Parameter	Description
Community	The name of the community for SNMPv1 and SNMPv2c. This acts like a weak password.

User

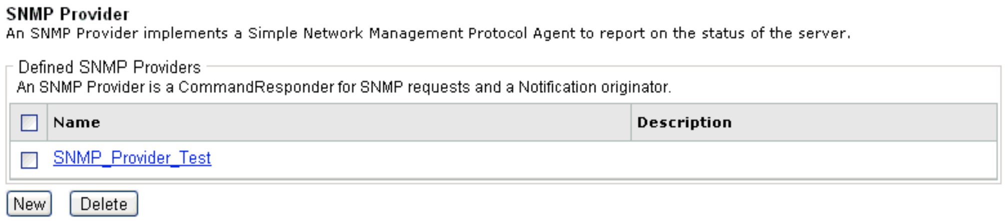
Parameter	Description
User Name	The name of the user to register in the User-based Security Model MIB for SNMPv3.
Authentication Protocol	The authentication protocol to use if authentication is enabled in the security level.
Authentication Passphrase	The authentication passphrase to use if authentication is enabled in the security level.
Privacy Protocol	The privacy protocol to use if privacy is enabled in the security level.
Privacy Passphrase	The privacy passphrase to use if privacy is enabled in the security level.

Notification

Parameter	Description
Send Notifications	Determines whether the server will send SNMP notifications.
Notification Processing Model	The message processing model used to send the notification.
Notification Type	The type of Protocol Data Unit used to send the notification. The Inform notification requires processing model v2c or v3.
Notification Protocol	The protocol over which the notification will be sent.
Notification Host	The host where the notification will be sent.
Notification Port	The port where the notification will be sent. The default is 162.
Notification Timeout	The time allocated to send the notification in 100th of a second.
Notification Retry Count	The number of attempts to send the notification. The default is 1.

4. Click *Add* when you are finished.

   You are returned to the main SNMP Provider pane and the new SNMP provider that was defined is added to the list.



SNMP Provider

An SNMP Provider implements a Simple Network Management Protocol Agent to report on the status of the server.

Defined SNMP Providers

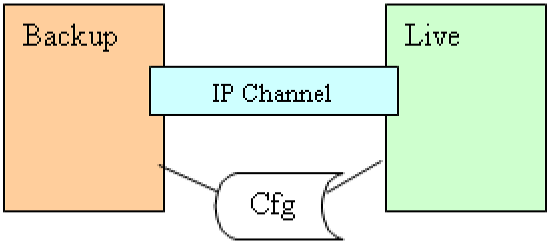
An SNMP Provider is a CommandResponder for SNMP requests and a Notification originator.

□	Name	Description
□	SNMP_Provider_Test	

5. To define multiple SNMP providers, repeat this procedure.

iWay Service Manager Failover

Failover enables a suspended iWay Service Manager (iSM) instance to "shadow" a live iSM instance. If the live iSM instance fails, then the suspended iSM instance leaves its suspended state and begins full operation. The servers can opt to share a configuration, allowing single-point management of the system. The following image illustrates this shared configuration.



The backup iSM instance is started with the command line switch -b or the Java system property iway.backup set to true. The default is live operation. The backup engine monitors the designated IP channel for activity initiated by the live system. If no activity is detected for a period of time, then the backup system enters live mode.

A hot backup system that goes live acts as a live system, using the backup location to begin to attempt a heartbeat.

Configuration

At start up, the failover (backup) system begins with the -b parameter (iwsrv -b). You can also run the server with the Java system property iway.backup=true. Either option instructs the server to begin as a backup server. You can configure properties using the Service Manager console.

The following table lists and describes the configuration properties.

Property	Description
Location of Backup **Note:** On live system	Location of live system failover partner. Each URL entry must carry an attribute of the name of the server to which it applies. Heartbeat signals are sent to the location for live systems. Location is in the form, host:port, for example, 1:8989.
Heartbeat Port **Note:** On backup system	Port on which the backup server listens for the live system heartbeat, for example, 8989.
Threshold **Note:** On backup system	Period to wait for a live signal.

The listener monitor shows the backup listeners in the state of WAITING.

Sequence of Operation

❏ Live System

The location is extracted on start up. Each live server can have only one backup server.

Periodically, the server sends a heartbeat signal to the *host:port* identified as the failover for the server.

When a stop command is entered, the stop signal is sent to the backup server to prevent it from becoming live.

- Backup System

  On startup, the backup system checks for the <hotbackup> configuration. It sets an accept operation on the backup port and awaits signals. After the first signal arrives, it begins a countdown clock for the threshold time period. One heartbeat signal is required to set the backup into failover mode; this allows the backup system to be started before the live system.

  As signals arrive, a timeout clock is reset. Should the time expire, the failover system enters live mode.

  As a best practice, the time-out threshold should never be less than three seconds, and preferably it should be at least five seconds.

  As each heartbeat arrives, it is checked to determine whether this is a stop signal. If so, the failover returns to initial mode to await a heartbeat signal to restart the cycle.

  When the system becomes live, it begins normal operation. A common practice is to point a file listener at a startup document that is emitted by email. This informs an administrator that a hot backup occurred.

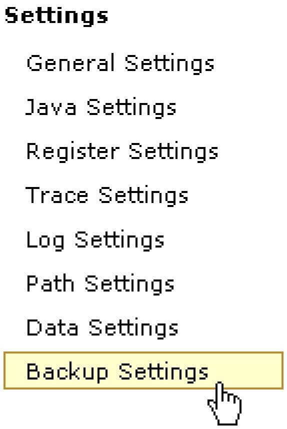
Caution: The engine alone cannot institute a complete hot backup capability. Protocols that carry only virtual names can be switched over. Others, especially the TCP-based protocols, cannot be switched over. Unless the hot back up is on the same computer as the failing system it takes over (thus voiding some of the purpose of hot back up), the field client must "know" the host name. Accordingly, customers must use commercial TCP switches to alleviate this issue.

This issue is less critical in cases where iWay Software is on both sides of the interface, as is the case with MQSI nodes. In this case, a retry for a second host address can be made part of the recovery cycle.

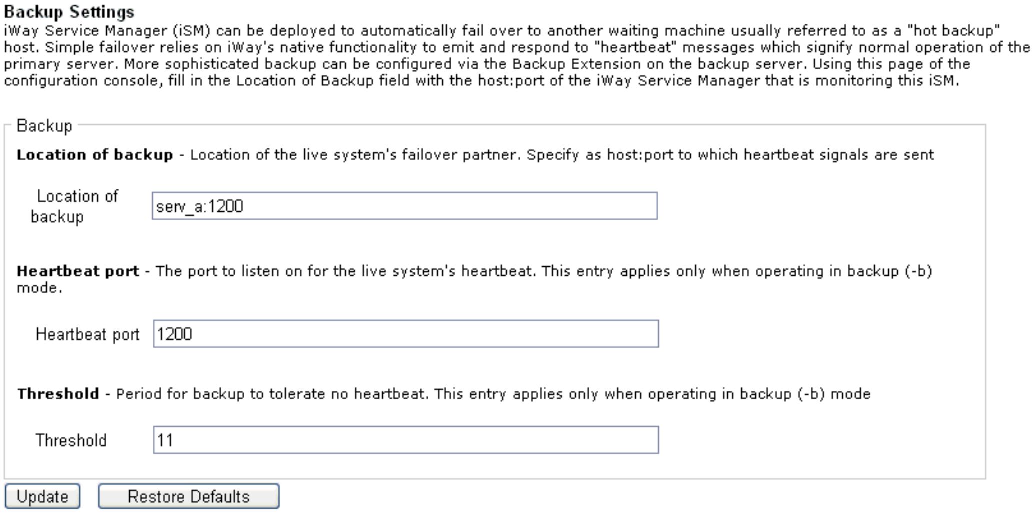
Both systems must be on the same side of the firewall, and the backup system must be reachable from the active system through TCP.

In the following example, the hot backup system is on server iam1, which is listening on port 1200. The live server uses this entry to determine where to send heartbeats. The backup server uses the heartbeat port entry to determine the port on which it is listening and the threshold field to determine the number of seconds to tolerate a loss of heartbeat before attempting to take over.

In the left pane of the iWay Service Manager Administration Console, click *Backup Settings*.



The Backup Settings pane is displayed, as shown in the following image.



Backup Settings

iWay Service Manager (iSM) can be deployed to automatically fail over to another waiting machine usually referred to as a "hot backup" host. Simple failover relies on iWay's native functionality to emit and respond to "heartbeat" messages which signify normal operation of the primary server. More sophisticated backup can be configured via the Backup Extension on the backup server. Using this page of the configuration console, fill in the Location of Backup field with the host:port of the iWay Service Manager that is monitoring this iSM.

Backup

Location of backup - Location of the live system's failover partner. Specify as host:port to which heartbeat signals are sent

Location of backup: serv_a:1200

Heartbeat port - The port to listen on for the live system's heartbeat. This entry applies only when operating in backup (-b) mode.

Heartbeat port: 1200

Threshold - Period for backup to tolerate no heartbeat. This entry applies only when operating in backup (-b) mode

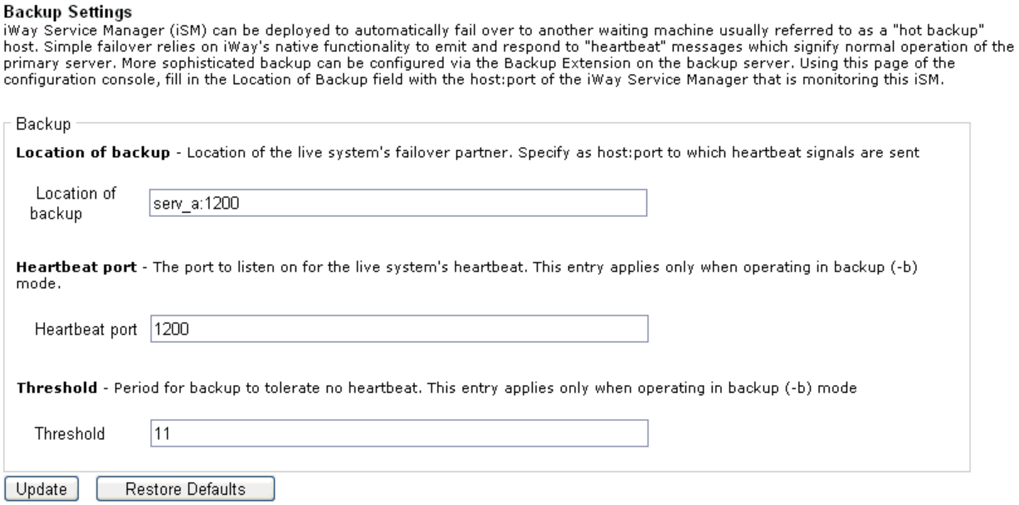
Threshold: 11

A single Service Manager cannot be both a live server and a backup server. The -b parameter is used to determine whether the server is a back up. Only the location of the backup field applies to live servers; the heartbeat port and threshold apply to the backup server. Because you can change a server from live to a back up depending upon how it is started, all fields are available.

iWay Software does not recommend a hot back up on the same system as the live server, as the hot back up is intended to compensate for the unexpected loss of a complete system. Usually, this is caused by loss of the computer itself and therefore, having the hot back up on the same system would result in the loss of both.

Example: Using Hot Backup

The following configuration works but is not recommended. The following example shows two servers, serv_a and serv_b. Each uses the other as a back up, using the same port. Usually, serv_a is the live server, and serv_b is the back up. Both use port 1200. The serv_a configuration is shown in the following image.



Backup Settings
iWay Service Manager (iSM) can be deployed to automatically fail over to another waiting machine usually referred to as a "hot backup" host. Simple failover relies on iWay's native functionality to emit and respond to "heartbeat" messages which signify normal operation of the primary server. More sophisticated backup can be configured via the Backup Extension on the backup server. Using this page of the configuration console, fill in the Location of Backup field with the host:port of the iWay Service Manager that is monitoring this iSM.

Backup

Location of backup - Location of the live system's failover partner. Specify as host:port to which heartbeat signals are sent

Location of backup: serv_a:1200

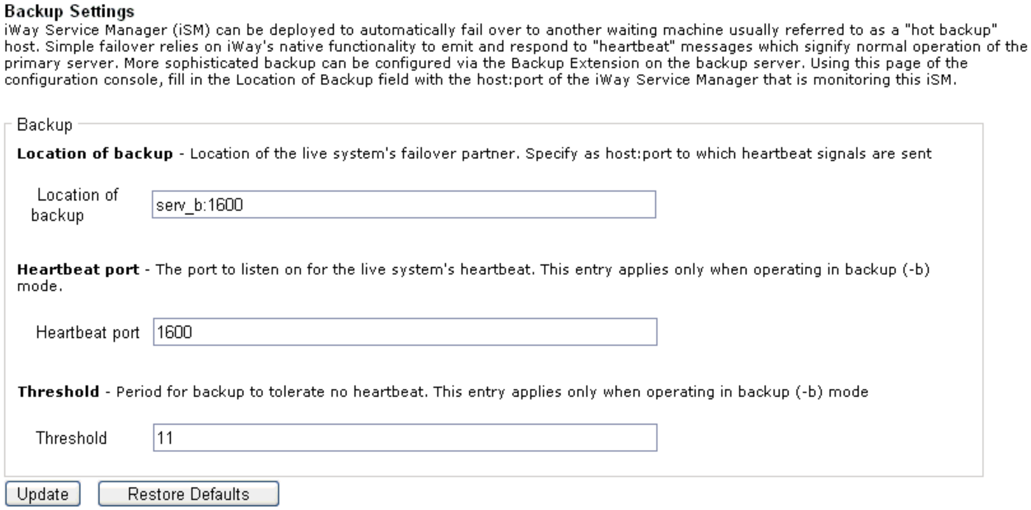
Heartbeat port - The port to listen on for the live system's heartbeat. This entry applies only when operating in backup (-b) mode.

Heartbeat port: 1200

Threshold - Period for backup to tolerate no heartbeat. This entry applies only when operating in backup (-b) mode

Threshold: 11

The serv_b configuration is shown in the following image:



Backup Settings

iWay Service Manager (iSM) can be deployed to automatically fail over to another waiting machine usually referred to as a "hot backup" host. Simple failover relies on iWay's native functionality to emit and respond to "heartbeat" messages which signify normal operation of the primary server. More sophisticated backup can be configured via the Backup Extension on the backup server. Using this page of the configuration console, fill in the Location of Backup field with the host:port of the iWay Service Manager that is monitoring this iSM.

Backup

Location of backup - Location of the live system's failover partner. Specify as host:port to which heartbeat signals are sent

Location of backup: serv_b:1600

Heartbeat port - The port to listen on for the live system's heartbeat. This entry applies only when operating in backup (-b) mode.

Heartbeat port: 1600

Threshold - Period for backup to tolerate no heartbeat. This entry applies only when operating in backup (-b) mode

Threshold: 11

You can start either server as the back up using the -b parameter.

Channel Management

Note: This section is being deprecated with the use of iWay Integration Tools (iIT). For more information, see the *iWay Integration Tools User's Guide*.

After you have built a channel using the iWay Service Manager Administration Console, you must deploy it to be used.

Deployment is the mechanism by which channels move from being stored in the Registry to becoming active in iWay Service Manager. Deployment separates the creation of iSM functions into a design phase and an implementation phase. Deploying and undeploying are useful during development and on a production server, since channels can evolve through multiple versions.

Deploying a Channel

Deploying a channel is the third stage in channel management. Deployment takes a built channel and deploys its components into a runtime environment. When you deploy a channel, you deploy a version of the built channel.

Procedure: How to Deploy a Channel

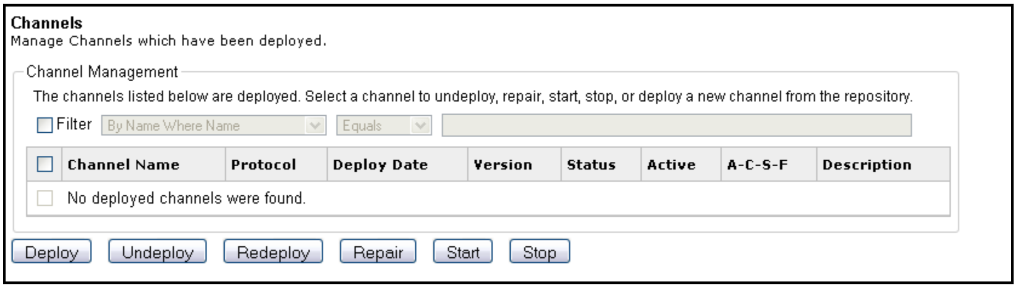
To deploy a channel:

1. Click *Deployments* in the menu bar.

   The Deployments pane opens, showing links to Channels, Services, and Web Services, in the left pane.

2. In the left console pane of the Deployments menu, select *Channels*.

   The Channel Management pane opens. This can also show already deployed channels.



Channels
Manage Channels which have been deployed.

Channel Management

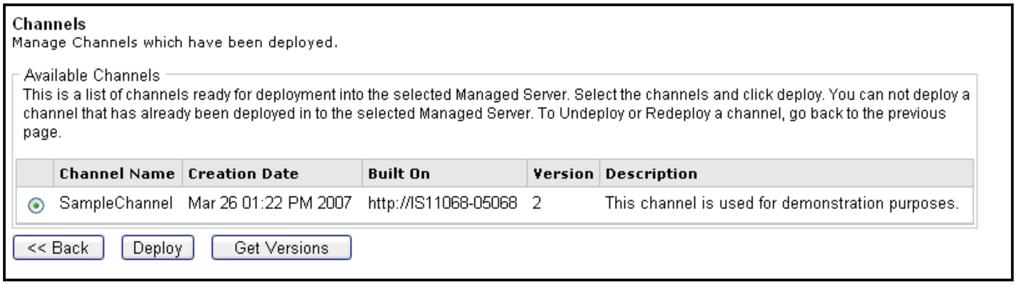
The channels listed below are deployed. Select a channel to undeploy, repair, start, stop, or deploy a new channel from the repository.

□ Filter By Name Where Name Equals

□	Channel Name	Protocol	Deploy Date	Version	Status	Active	A-C-S-F	Description
□	No deployed channels were found.							

3. Click *Deploy*.

   The Available Channels pane opens.



Channels
Manage Channels which have been deployed.

Available Channels

This is a list of channels ready for deployment into the selected Managed Server. Select the channels and click deploy. You can not deploy a channel that has already been deployed in to the selected Managed Server. To Undeploy or Redeploy a channel, go back to the previous page.

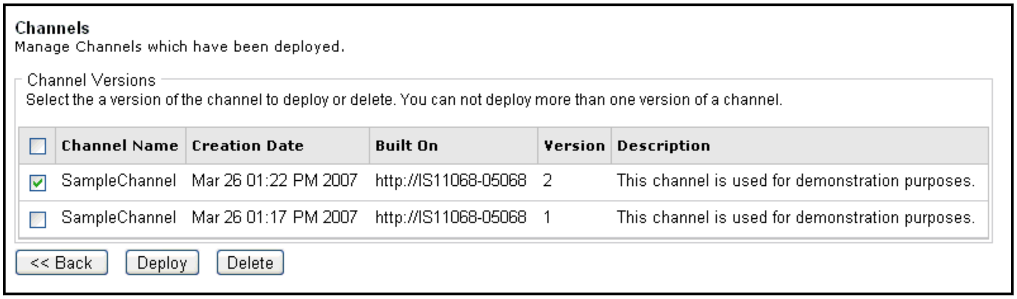
	Channel Name	Creation Date	Built On	Version	Description
●	SampleChannel	Mar 26 01:22 PM 2007	http://IS11068-05068	2	This channel is used for demonstration purposes.

The table that is provided lists all channels that have successfully completed the build process. It also includes the channel creation date, the system where the channel was created, a version number, and a short description.

If there are multiple versions of a channel, you must deploy a specific version of the channel. In this example, there are two versions of SampleChannel.

4. Click *Get Versions*.

The Channel Versions pane opens.



Channels
Manage Channels which have been deployed.

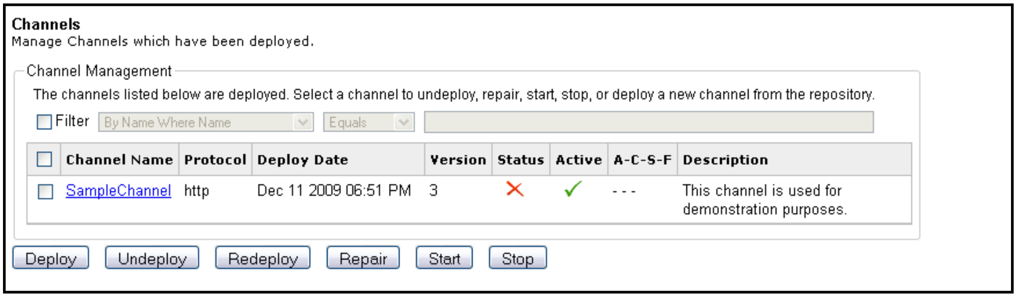
Channel Versions
Select the a version of the channel to deploy or delete. You can not deploy more than one version of a channel.

□	Channel Name	Creation Date	Built On	Version	Description
☑	SampleChannel	Mar 26 01:22 PM 2007	http://IS11068-05068	2	This channel is used for demonstration purposes.
□	SampleChannel	Mar 26 01:17 PM 2007	http://IS11068-05068	1	This channel is used for demonstration purposes.

Creating multiple versions of a channel is useful if the current version is not working properly and you want to revert back to a previous version that worked.

5. Select the channel version you want to deploy, for example, SampleChannel, Version 2, and click *Deploy*.

   You are returned to the Channel Management pane. Notice that SampleChannel is now included in the list of deployed channels.



Channels
Manage Channels which have been deployed.

Channel Management
The channels listed below are deployed. Select a channel to undeploy, repair, start, stop, or deploy a new channel from the repository.

□ Filter By Name Where Name Equals

□	Channel Name	Protocol	Deploy Date	Version	Status	Active	A-C-S-F	Description
□	SampleChannel	http	Dec 11 2009 06:51 PM	3	✗	✓	- - -	This channel is used for demonstration purposes.

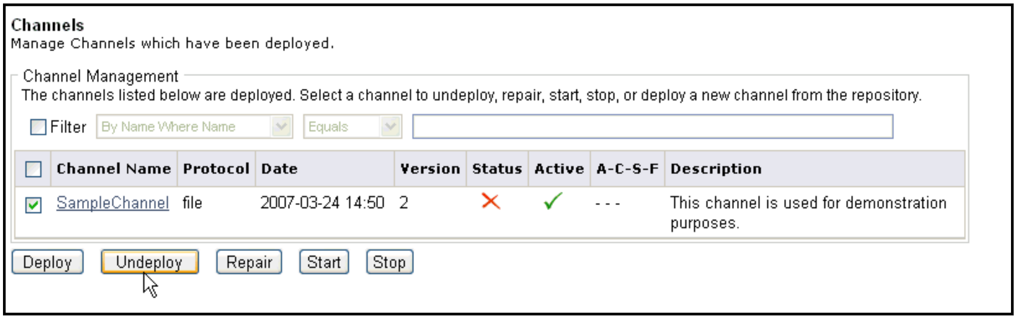
Once a channel is deployed, it must be started. For more information, see *How to Start a Channel* on page 443.

Procedure: How to Undeploy a Channel

To undeploy a channel:

1. In the left console pane of the Deployments menu, select *Channels*.

The Channel Management pane opens.



Channels
Manage Channels which have been deployed.

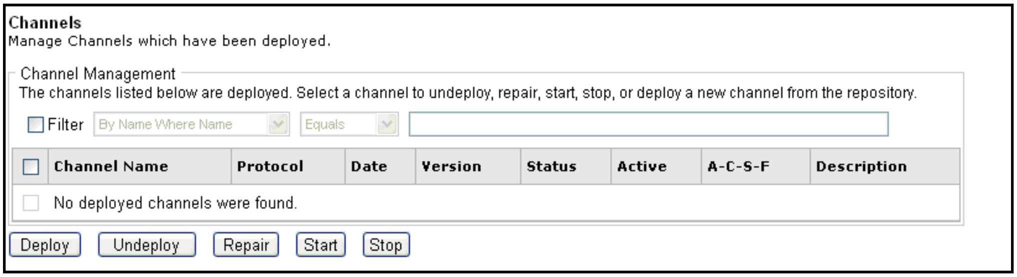
Channel Management
The channels listed below are deployed. Select a channel to undeploy, repair, start, stop, or deploy a new channel from the repository.

□ Filter By Name Where Name | Equals

□	Channel Name	Protocol	Date	Version	Status	Active	A-C-S-F	Description
☑	SampleChannel	file	2007-03-24 14:50	2	✗	✓	- - -	This channel is used for demonstration purposes.

2. Select the channel you want to undeploy, for example, SampleChannel, and click *Undeploy*.

   The channel is undeployed and no longer appears in the Channel Management pane.



Channels
Manage Channels which have been deployed.

Channel Management
The channels listed below are deployed. Select a channel to undeploy, repair, start, stop, or deploy a new channel from the repository.

□ Filter By Name Where Name | Equals

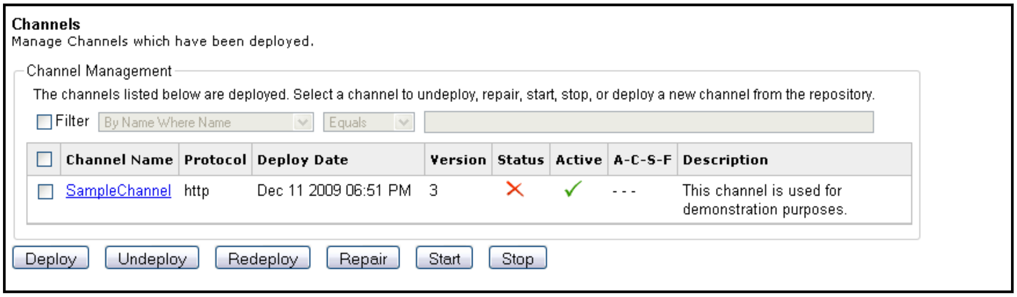
□	Channel Name	Protocol	Date	Version	Status	Active	A-C-S-F	Description
□	No deployed channels were found.							

Procedure: How to Start a Channel

To start a channel:

1. In the left console pane of the Deployments menu, select *Channels*.

   The Channel Management pane opens.



Channels
Manage Channels which have been deployed.

Channel Management
The channels listed below are deployed. Select a channel to undeploy, repair, start, stop, or deploy a new channel from the repository.

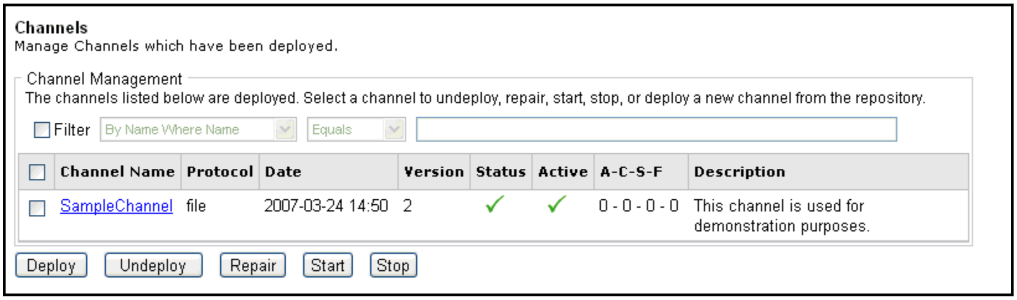
□ Filter By Name Where Name | Equals

□	Channel Name	Protocol	Deploy Date	Version	Status	Active	A-C-S-F	Description
□	SampleChannel	http	Dec 11 2009 06:51 PM	3	✗	✓	- - -	This channel is used for demonstration purposes.

The red X in the Status column indicates that a channel is not started.

2. Select the check box next to a deployed channel you wish to start, for example, SampleChannel.
3. Click *Start*.

   A green check mark now displays in the Status column, indicating that the deployed channel is started.



Channels
Manage Channels which have been deployed.

Channel Management
The channels listed below are deployed. Select a channel to undeploy, repair, start, stop, or deploy a new channel from the repository.

□ Filter By Name Where Name Equals

□	Channel Name	Protocol	Date	Version	Status	Active	A-C-S-F	Description
□	SampleChannel	file	2007-03-24 14:50	2	✓	✓	0 - 0 - 0 - 0	This channel is used for demonstration purposes.

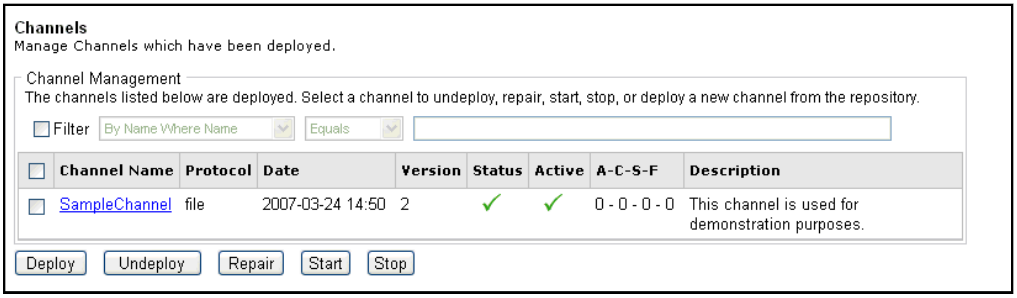
Tip: To start more than one channel at once, you can select multiple check boxes in the Channel Management pane and click *Start*.

Procedure: How to Stop a Channel

To stop a channel:

1. In the left console pane of the Deployments menu, select *Channels*.

   The Channel Management pane opens.



Channels
Manage Channels which have been deployed.

Channel Management
The channels listed below are deployed. Select a channel to undeploy, repair, start, stop, or deploy a new channel from the repository.

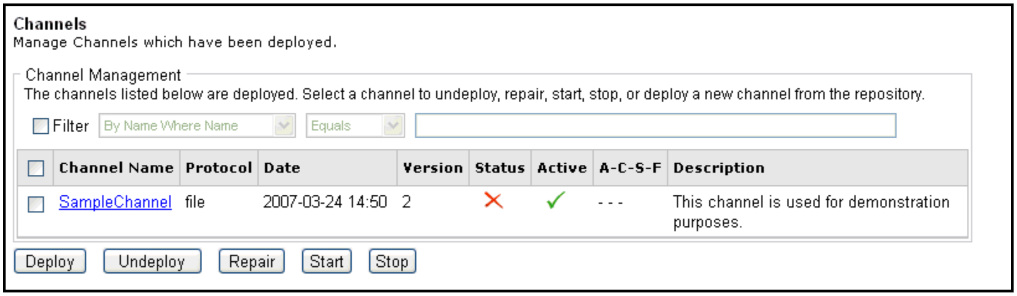
□ Filter By Name Where Name Equals

□	Channel Name	Protocol	Date	Version	Status	Active	A-C-S-F	Description
□	SampleChannel	file	2007-03-24 14:50	2	✓	✓	0 - 0 - 0 - 0	This channel is used for demonstration purposes.

   The green check mark in the status column indicates that a channel is started.

2. Select the check box next to a deployed channel you wish to stop, for example, SampleChannel.
3. Click *Stop*.

A red X now displays in the Status column, indicating that the deployed channel is stopped.



Channels

Manage Channels which have been deployed.

Channel Management

The channels listed below are deployed. Select a channel to undeploy, repair, start, stop, or deploy a new channel from the repository.

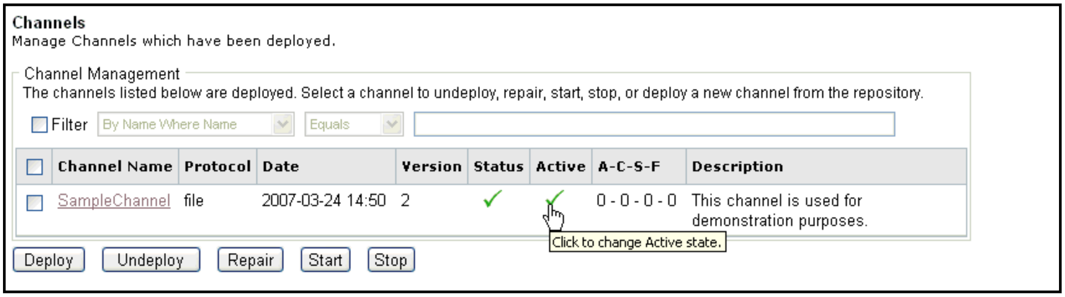
□	Channel Name	Protocol	Date	Version	Status	Active	A-C-S-F	Description
□	SampleChannel	file	2007-03-24 14:50	2	✗	✓	- - -	This channel is used for demonstration purposes.

Tip: To stop more than one channel at once, you can select multiple check boxes in the Channel Management pane and click *Stop*.

Procedure: How to Activate and Deactivate a Channel

Active means that the channel will start automatically during the next restart of iWay Service Manager. If you do not want the channel to start automatically during the next restart of iSM, you must deactivate the channel.

To make a channel active, click the red X in the Active column. To make a channel inactive, click on the green check mark in the Active column. In either case, the Channels pane will refresh and display the changed status in the Active column. Unlike starting and stopping channels, you may only toggle the Active status one channel at a time.



Channels

Manage Channels which have been deployed.

Channel Management

The channels listed below are deployed. Select a channel to undeploy, repair, start, stop, or deploy a new channel from the repository.

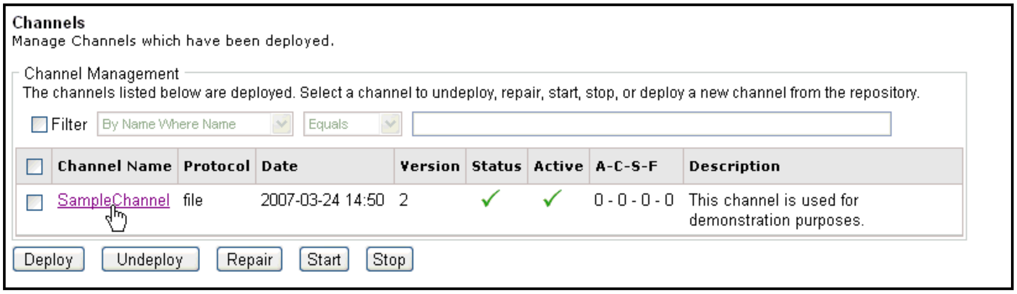
□	Channel Name	Protocol	Date	Version	Status	Active	A-C-S-F	Description
□	SampleChannel	file	2007-03-24 14:50	2	✓	✓	0 - 0 - 0 - 0	This channel is used for demonstration purposes.

Procedure: How to View Channel Traces

To view the trace log associated with a channel:

1. In the left console pane of the Deployments menu, select *Channels*.

The Channel Management pane opens.



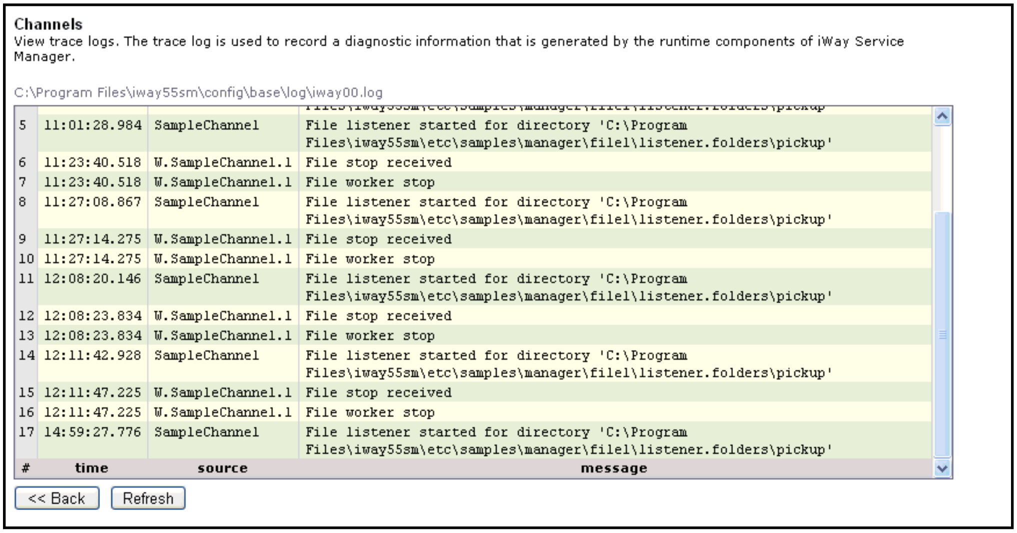
Channels
Manage Channels which have been deployed.

Channel Management
The channels listed below are deployed. Select a channel to undeploy, repair, start, stop, or deploy a new channel from the repository.

☐	Channel Name	Protocol	Date	Version	Status	Active	A-C-S-F	Description
☐	SampleChannel	file	2007-03-24 14:50	2	✓	✓	0 - 0 - 0 - 0	This channel is used for demonstration purposes.

2. Click the name of a channel, for example, SampleChannel.

   The trace log for the selected channel opens.



Channels
View trace logs. The trace log is used to record a diagnostic information that is generated by the runtime components of iWay Service Manager.

C:\Program Files\iway55sm\config\base\log\iway00.log

```
                                 Files\iway55sm\etc\samples\manager\file1\listener.folders\pickup'
5  11:01:28.984 SampleChannel    File listener started for directory 'C:\Program
                                 Files\iway55sm\etc\samples\manager\file1\listener.folders\pickup'
6  11:23:40.518 W.SampleChannel.1 File stop received
7  11:23:40.518 W.SampleChannel.1 File worker stop
8  11:27:08.867 SampleChannel    File listener started for directory 'C:\Program
                                 Files\iway55sm\etc\samples\manager\file1\listener.folders\pickup'
9  11:27:14.275 W.SampleChannel.1 File stop received
10 11:27:14.275 W.SampleChannel.1 File worker stop
11 12:08:20.146 SampleChannel    File listener started for directory 'C:\Program
                                 Files\iway55sm\etc\samples\manager\file1\listener.folders\pickup'
12 12:08:23.834 W.SampleChannel.1 File stop received
13 12:08:23.834 W.SampleChannel.1 File worker stop
14 12:11:42.928 SampleChannel    File listener started for directory 'C:\Program
                                 Files\iway55sm\etc\samples\manager\file1\listener.folders\pickup'
15 12:11:47.225 W.SampleChannel.1 File stop received
16 12:11:47.225 W.SampleChannel.1 File worker stop
17 14:59:27.776 SampleChannel    File listener started for directory 'C:\Program
                                 Files\iway55sm\etc\samples\manager\file1\listener.folders\pickup'
#      time         source                              message
```

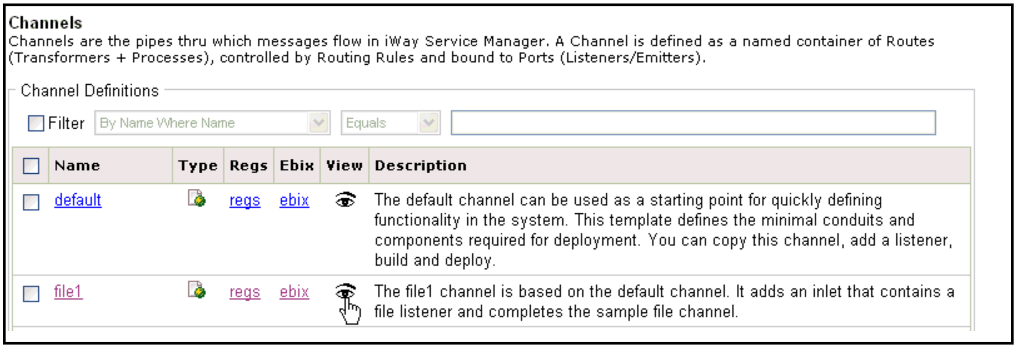
3. Click *Refresh* to view the latest instance of the channel trace log.

Procedure: How to Visualize a Channel

To visualize a channel:

1. In the left console pane of the Registry menu, select *Channels*.

The Channels pane opens.



Channels
Channels are the pipes thru which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

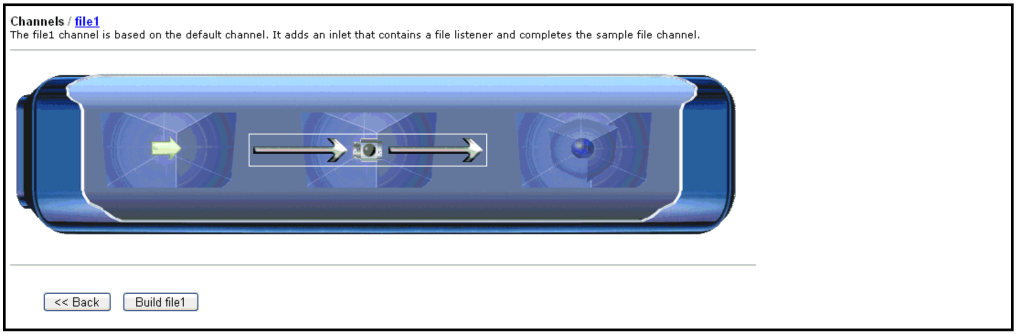
Channel Definitions

□ Filter By Name Where Name Equals

□	Name	Type	Regs	Ebix	View	Description
□	default		regs	ebix		The default channel can be used as a starting point for quickly defining functionality in the system. This template defines the minimal conduits and components required for deployment. You can copy this channel, add a listener, build and deploy.
□	file1		regs	ebix		The file1 channel is based on the default channel. It adds an inlet that contains a file listener and completes the sample file channel.

2. In the View column, click the eye icon for the channel you want to view, for example, file1.

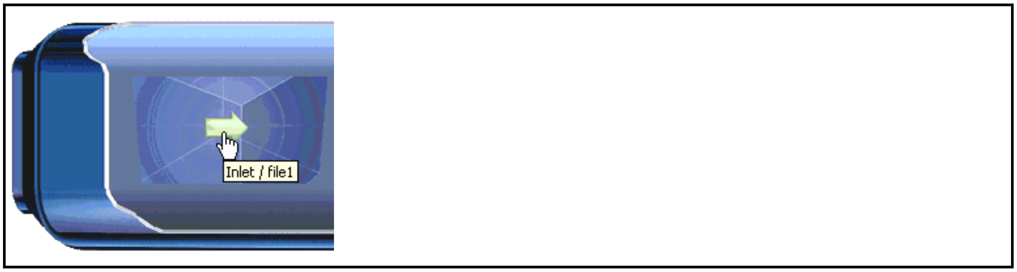
   A graphical representation of the channel, including the registered channel components, is displayed.



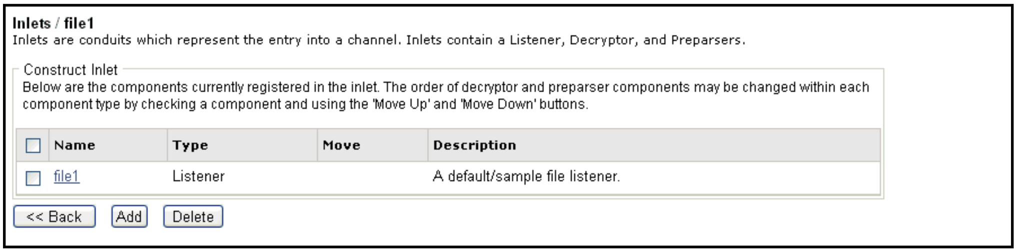
Channels / file1
The file1 channel is based on the default channel. It adds an inlet that contains a file listener and completes the sample file channel.

The left portion of the image represents the inlet, the middle portion represents the route, and the right portion represents the outlet.

If you move your mouse cursor over any of the portions in the channel image map, the cursor changes to a hand selection icon. The name of the specific channel component is provided.



These portions of the image map are active areas that allow you to navigate to the channel component that is defined in the Registry. For example, if you click the *Inlet / file1* portion of the image, the following pane opens:



Inlets / **file1**

Inlets are conduits which represent the entry into a channel. Inlets contain a Listener, Decryptor, and Preparsers.

Construct Inlet

Below are the components currently registered in the inlet. The order of decryptor and preparser components may be changed within each component type by checking a component and using the 'Move Up' and 'Move Down' buttons.

□	Name	Type	Move	Description
□	file1	Listener		A default/sample file listener.

3. Click *Back* to return to the Channels pane.

   After making any changes to the channel components, click *Build* to rebuild the channel.

Repairing a Channel

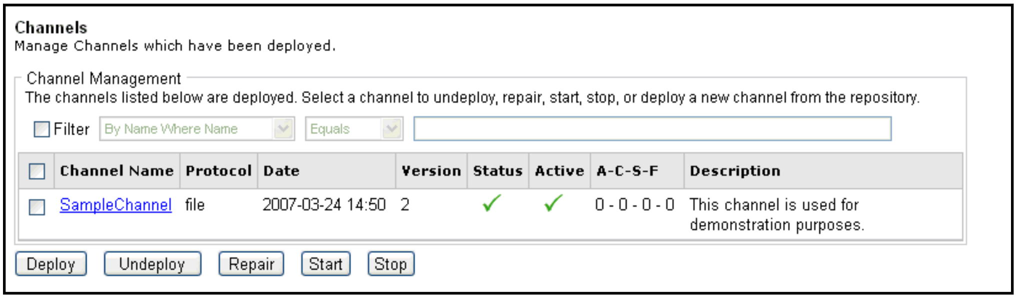
Repairing a channel allows you to refresh the current version of a channel from the last deployed Channel Archive (CAR) file.

Procedure: How to Repair a Channel

To repair a channel:

1. In the left console pane of the Deployments menu, select *Channels*.

   The Channel Management pane opens.



Channels

Manage Channels which have been deployed.

Channel Management

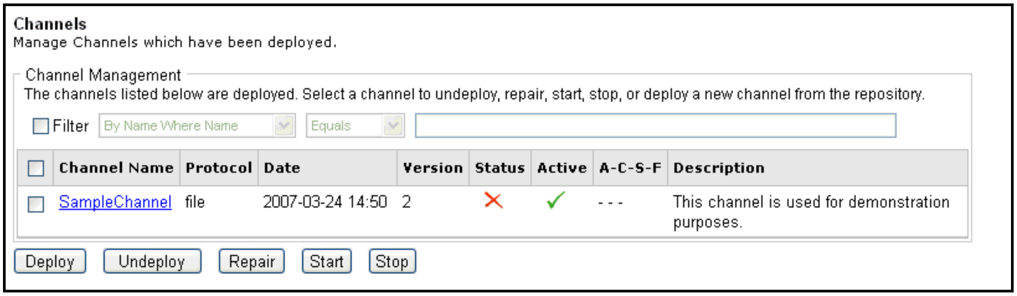
The channels listed below are deployed. Select a channel to undeploy, repair, start, stop, or deploy a new channel from the repository.

□	Channel Name	Protocol	Date	Version	Status	Active	A-C-S-F	Description
□	SampleChannel	file	2007-03-24 14:50	2	✓	✓	0 - 0 - 0 - 0	This channel is used for demonstration purposes.

2. Select the check box next to a deployed channel you wish to repair, for example, SampleChannel.

3. Click *Repair*.

The repaired channel is left in the stopped state.



Channels

Manage Channels which have been deployed.

Channel Management

The channels listed below are deployed. Select a channel to undeploy, repair, start, stop, or deploy a new channel from the repository.

□	Channel Name	Protocol	Date	Version	Status	Active	A-C-S-F	Description
□	SampleChannel	file	2007-03-24 14:50	2	✗	✓	- - -	This channel is used for demonstration purposes.

Tip: To repair more than one channel at once, you can select multiple check boxes in the Channel Management pane and click *Repair*.

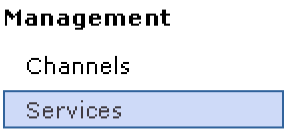
Service Management

Note: This section is being deprecated with the use of iWay Integration Tools (iIT). For more information, see the *iWay Integration Tools User's Guide*.

Deployment is the mechanism by which services move from being stored in the Registry to becoming active in iWay Service Manager (iSM). Deployment separates the creation of iSM functions into a design phase and an implementation phase. Deploying a service enables it to be used in an iSM run-time environment at a system level, not a listener level. Deployed services can also be exposed as web services if there are XML schemas at both ends (request/response). Deploying and undeploying are useful during development and on a production server, since services can evolve through multiple versions. For information on how to deploy a service as a web service, see the *iWay Designer User's Guide*.

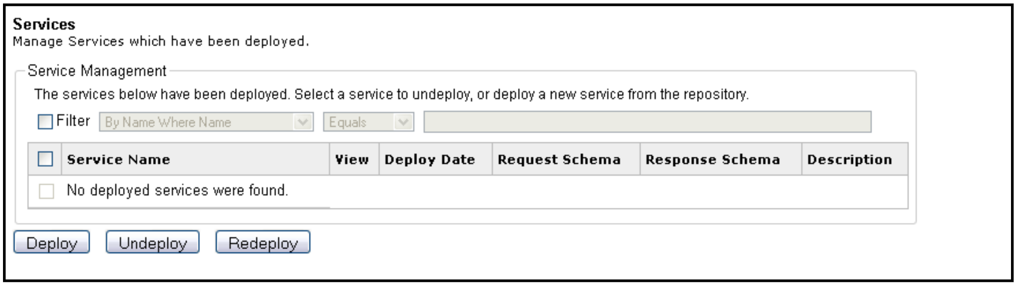
Procedure: How to Deploy a Service

To deploy a service:



1. In the left console pane of the Deployments menu, select *Services*.

The Service Management pane opens.



Services
Manage Services which have been deployed.

Service Management

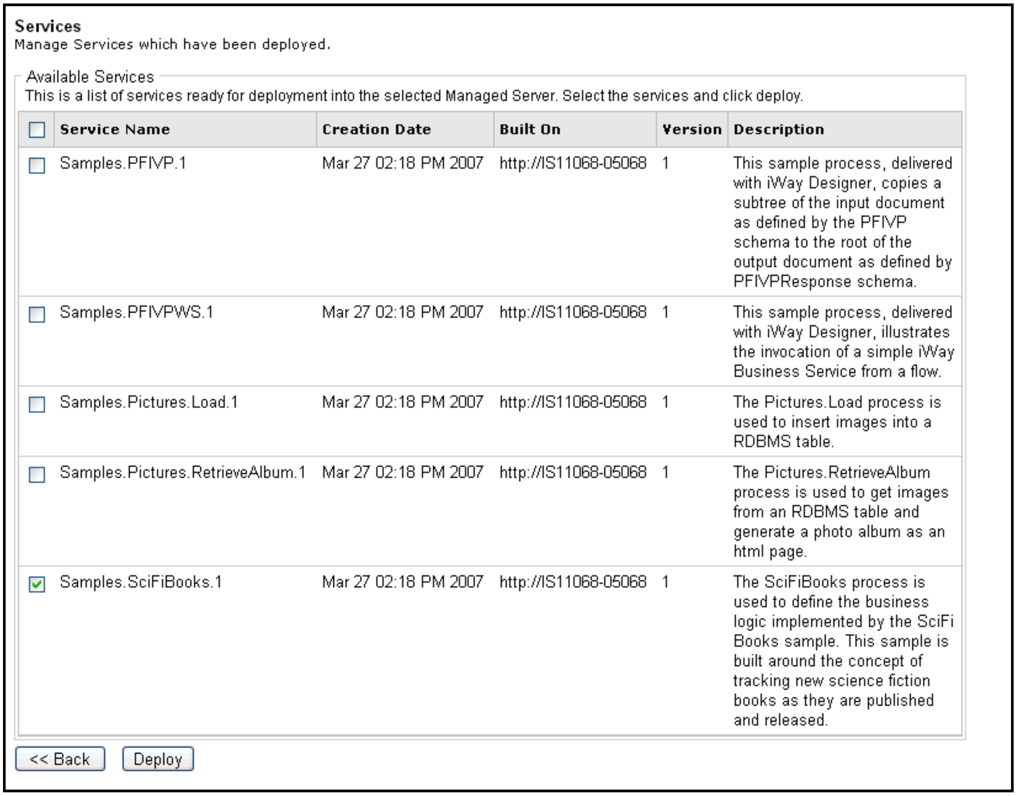
The services below have been deployed. Select a service to undeploy, or deploy a new service from the repository.

□ Filter By Name Where Name Equals

□	Service Name	View	Deploy Date	Request Schema	Response Schema	Description
□	No deployed services were found.					

2. Click *Deploy*.

   The Available Services pane opens.



Services
Manage Services which have been deployed.

Available Services

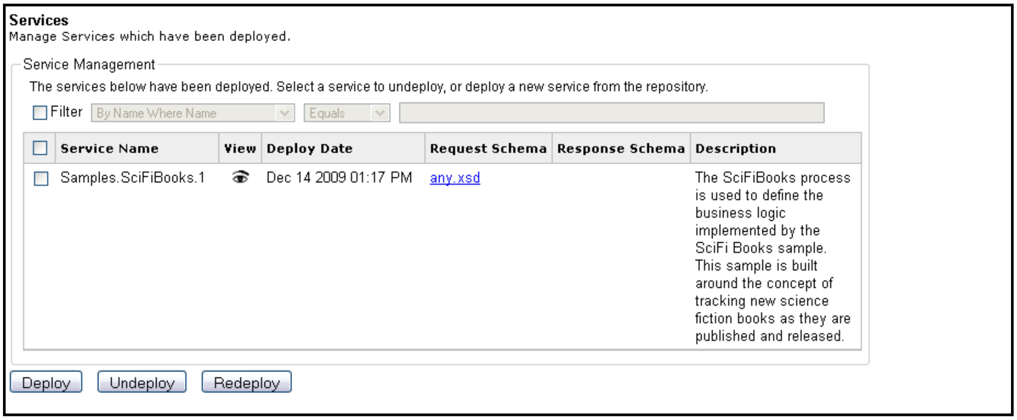
This is a list of services ready for deployment into the selected Managed Server. Select the services and click deploy.

□	Service Name	Creation Date	Built On	Version	Description
□	Samples.PFIVP.1	Mar 27 02:18 PM 2007	http://IS11068-05068	1	This sample process, delivered with iWay Designer, copies a subtree of the input document as defined by the PFIVP schema to the root of the output document as defined by PFIVPResponse schema.
□	Samples.PFIVPWS.1	Mar 27 02:18 PM 2007	http://IS11068-05068	1	This sample process, delivered with iWay Designer, illustrates the invocation of a simple iWay Business Service from a flow.
□	Samples.Pictures.Load.1	Mar 27 02:18 PM 2007	http://IS11068-05068	1	The Pictures.Load process is used to insert images into a RDBMS table.
□	Samples.Pictures.RetrieveAlbum.1	Mar 27 02:18 PM 2007	http://IS11068-05068	1	The Pictures.RetrieveAlbum process is used to get images from an RDBMS table and generate a photo album as an html page.
☑	Samples.SciFiBooks.1	Mar 27 02:18 PM 2007	http://IS11068-05068	1	The SciFiBooks process is used to define the business logic implemented by the SciFi Books sample. This sample is built around the concept of tracking new science fiction books as they are published and released.

The table that is provided lists all services that are ready for deployment. It also includes the service creation date, the system where the service was created, a version number, and a short description.

3. Select the service you want to deploy, for example, Samples.SciFiBooks.1, and click *Deploy*.

   You are returned to the Service Management pane. Notice that Samples.SciFiBooks.1 is now included in the list of deployed services and can be called by other process flows or deployed as a web service.



Services

Manage Services which have been deployed.

Service Management

The services below have been deployed. Select a service to undeploy, or deploy a new service from the repository.

□ Filter By Name Where Name Equals

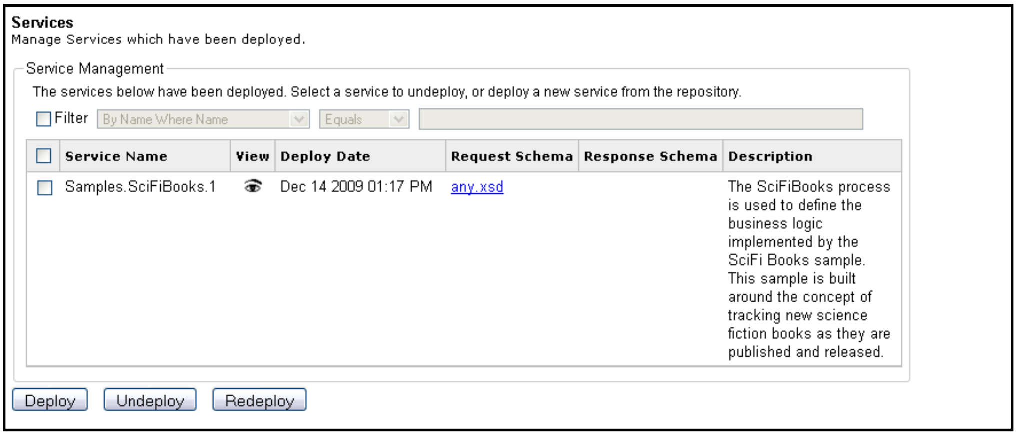
□	Service Name	View	Deploy Date	Request Schema	Response Schema	Description
□	Samples.SciFiBooks.1	👁	Dec 14 2009 01:17 PM	any.xsd		The SciFiBooks process is used to define the business logic implemented by the SciFi Books sample. This sample is built around the concept of tracking new science fiction books as they are published and released.

Procedure: How to Undeploy a Service

To undeploy a service:

1. In the left console pane of the Deployments menu, select *Services*.

   The Service Management pane opens.



Services

Manage Services which have been deployed.

Service Management

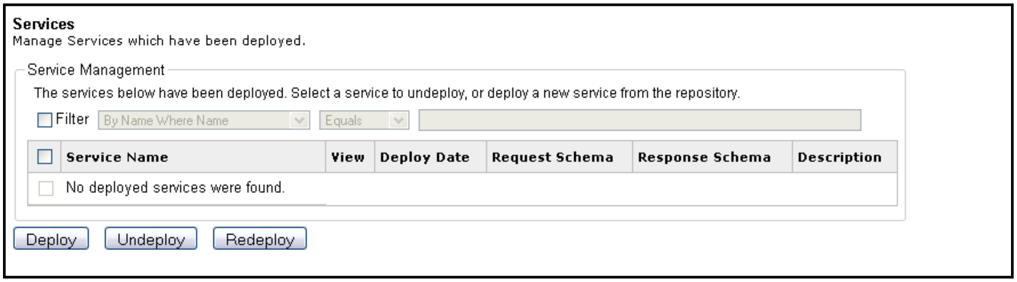
The services below have been deployed. Select a service to undeploy, or deploy a new service from the repository.

□ Filter By Name Where Name Equals

□	Service Name	View	Deploy Date	Request Schema	Response Schema	Description
□	Samples.SciFiBooks.1	👁	Dec 14 2009 01:17 PM	any.xsd		The SciFiBooks process is used to define the business logic implemented by the SciFi Books sample. This sample is built around the concept of tracking new science fiction books as they are published and released.

2. Select the service you want to undeploy, for example, Samples.SciFiBooks.1, and click *Undeploy*.

The service is undeployed and no longer appears in the Service Management pane.



Services

Manage Services which have been deployed.

Service Management

The services below have been deployed. Select a service to undeploy, or deploy a new service from the repository.

☐ Filter By Name Where Name Equals

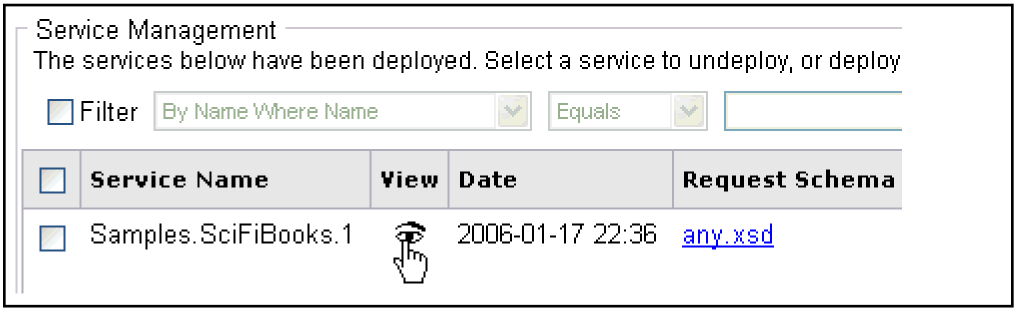
☐	Service Name	View	Deploy Date	Request Schema	Response Schema	Description
☐	No deployed services were found.					

Procedure: How to Visualize a Service

To visualize a service:

1. In the left console pane of the Deployments menu, select *Services*.

   The Service Management pane opens.



Service Management

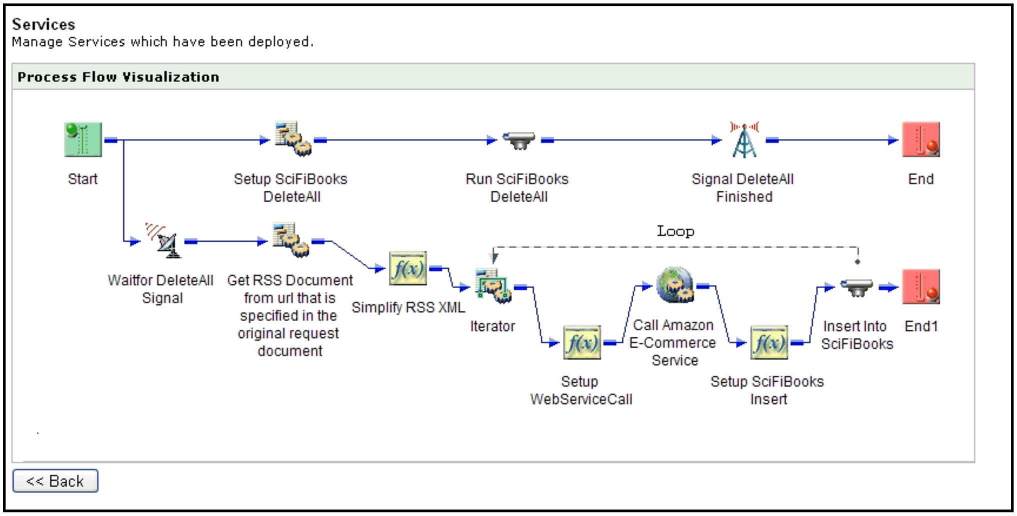
The services below have been deployed. Select a service to undeploy, or deploy

☐ Filter By Name Where Name Equals

☐	Service Name	View	Date	Request Schema
☐	Samples.SciFiBooks.1	👁	2006-01-17 22:36	any.xsd

2. Click the eye icon in the View column.

A graphical representation of the process flow that is encapsulated in the service is displayed.


Services
Manage Services which have been deployed.
Process Flow Visualization
Start
Setup SciFiBooks DeleteAll
Run SciFiBooks DeleteAll
Signal DeleteAll Finished
End
Loop
Waitfor DeleteAll Signal
Get RSS Document from url that is specified in the original request document
Simplify RSS XML
Iterator
Setup WebServiceCall
Call Amazon E-Commerce Service
Setup SciFiBooks Insert
Insert Into SciFiBooks
End1
<< Back

For more information on the structure of process flows, see the *iWay Designer User's Guide*.

3. Click *Back* to return to the Service Management pane.

Testing Web Services

Web services are created using the iWay Explorer, which is available as part of the iWay Designer (and in other forms). The console supports displaying and testing those web services.

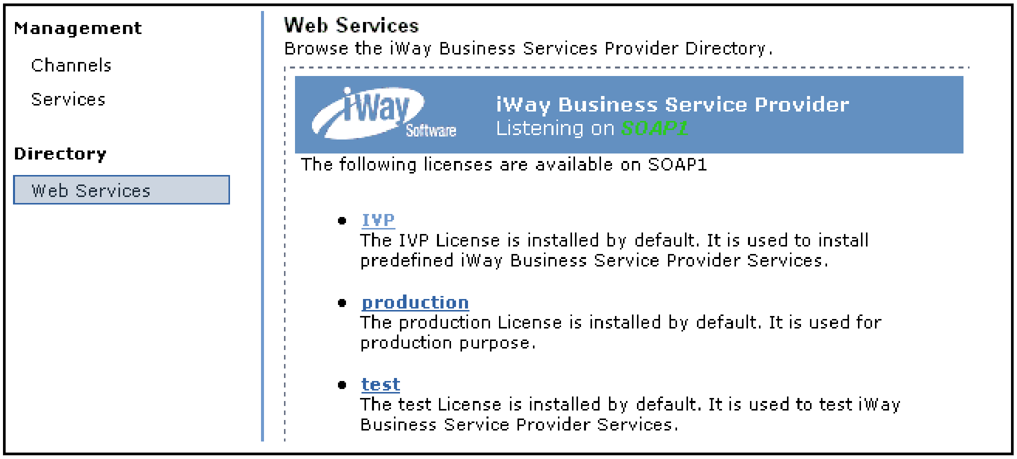
The Web Services link in the left console pane of the Deployments menu allows you to browse the directory of the iWay Business Service Provider (iBSP).

Procedure: How to Browse Web Services

To browse web services:

1. In the left console pane, click *Web Services*.

The Web Services pane opens.



Web Services

Browse the iWay Business Services Provider Directory.

iWay Business Service Provider
Listening on SOAP1

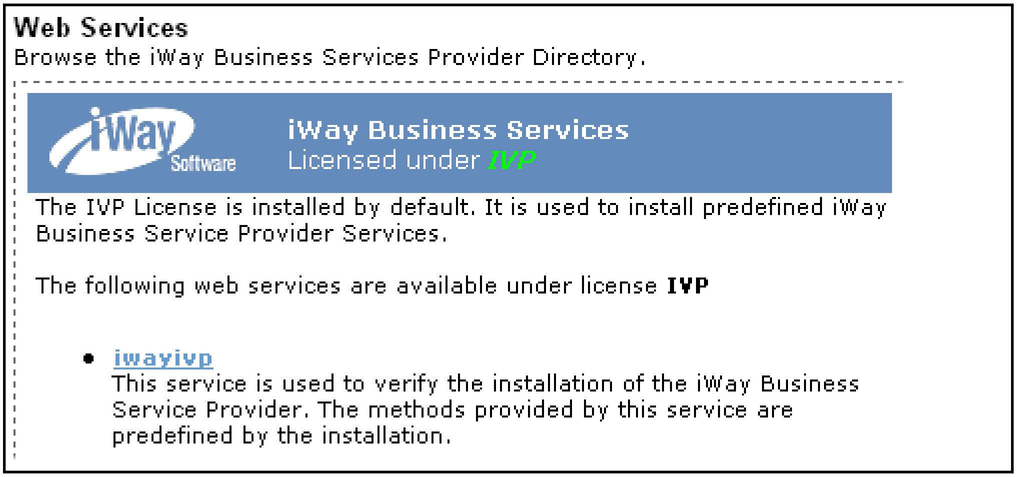
The following licenses are available on SOAP1

- **IVP**
  The IVP License is installed by default. It is used to install predefined iWay Business Service Provider Services.
- **production**
  The production License is installed by default. It is used for production purpose.
- **test**
  The test License is installed by default. It is used to test iWay Business Service Provider Services.

A list of available license names with short descriptions is provided.

2. Click the name of a specific license, for example, IVP.

   The available web services for the IVP license are displayed.



Web Services

Browse the iWay Business Services Provider Directory.

iWay Business Services
Licensed under IVP

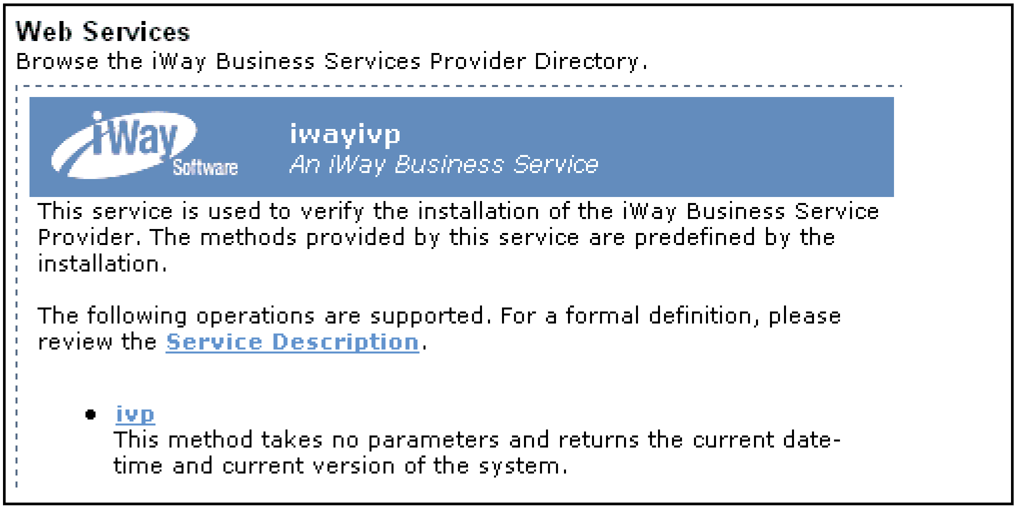
The IVP License is installed by default. It is used to install predefined iWay Business Service Provider Services.

The following web services are available under license **IVP**

- **iwayivp**
  This service is used to verify the installation of the iWay Business Service Provider. The methods provided by this service are predefined by the installation.

3. Click the name of the web service, for example, iwayivp.

The available methods for the iwayivp web service are displayed.



Web Services

Browse the iWay Business Services Provider Directory.

iwayivp

An iWay Business Service

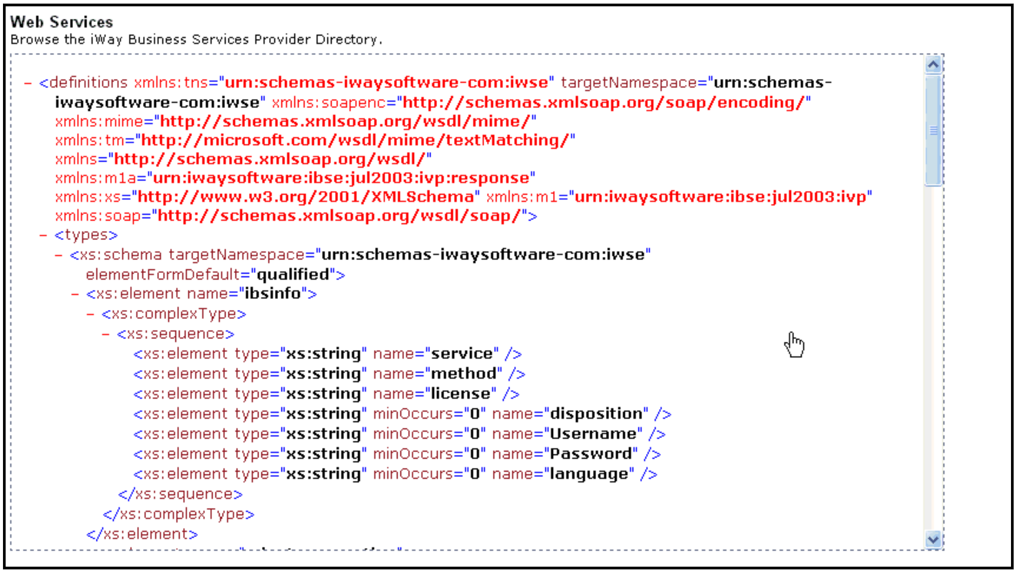
This service is used to verify the installation of the iWay Business Service Provider. The methods provided by this service are predefined by the installation.

The following operations are supported. For a formal definition, please review the Service Description.

- ivp
  This method takes no parameters and returns the current date-time and current version of the system.

4. Click the *Service Description* link to view the WSDL for the iwayivp web service.

   The following WSDL is displayed.



Web Services

Browse the iWay Business Services Provider Directory.

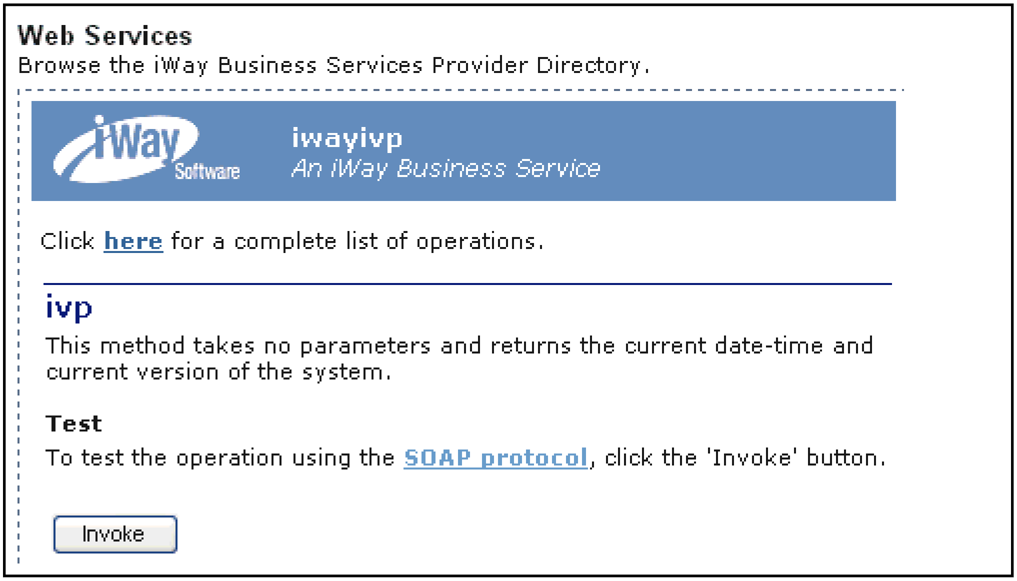
```
- <definitions xmlns:tns="urn:schemas-iwaysoftware-com:iwse" targetNamespace="urn:schemas-
    iwaysoftware-com:iwse" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
    xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:m1a="urn:iwaysoftware:ibse:jul2003:ivp:response"
    xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:m1="urn:iwaysoftware:ibse:jul2003:ivp"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
  - <types>
    - <xs:schema targetNamespace="urn:schemas-iwaysoftware-com:iwse"
        elementFormDefault="qualified">
      - <xs:element name="ibsinfo">
        - <xs:complexType>
          - <xs:sequence>
              <xs:element type="xs:string" name="service" />
              <xs:element type="xs:string" name="method" />
              <xs:element type="xs:string" name="license" />
              <xs:element type="xs:string" minOccurs="0" name="disposition" />
              <xs:element type="xs:string" minOccurs="0" name="Username" />
              <xs:element type="xs:string" minOccurs="0" name="Password" />
              <xs:element type="xs:string" minOccurs="0" name="language" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
```

5. Click the *Back* button on your browser to return to the list of available methods for the iwayivp web service.

   The names of the methods are links that can be used to invoke a method.

6. Click the name of a method, for example, ivp.

The Test pane opens.



Web Services

Browse the iWay Business Services Provider Directory.

iwayivp

An iWay Business Service

Click here for a complete list of operations.

ivp

This method takes no parameters and returns the current date-time and current version of the system.

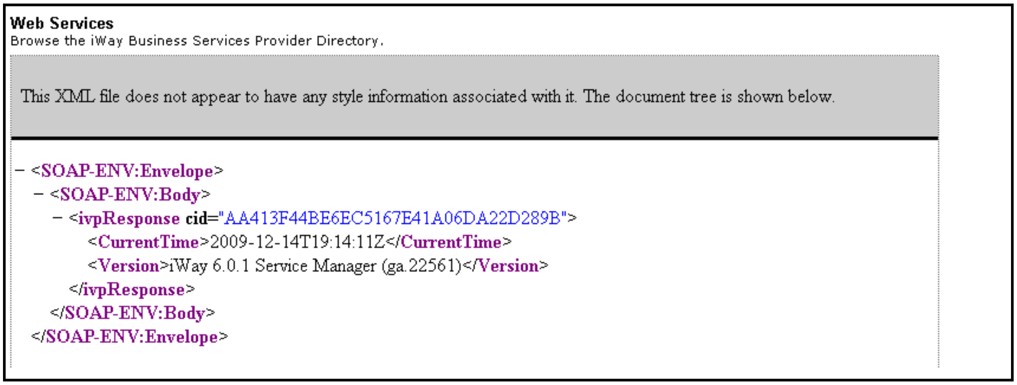
Test

To test the operation using the SOAP protocol, click the 'Invoke' button.

If the method requires input XML, a text box will be displayed. If input XML is not required, an Invoke button is displayed.

7. Click *Invoke*.

   The output of the web service is displayed for review.



Web Services

Browse the iWay Business Services Provider Directory.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

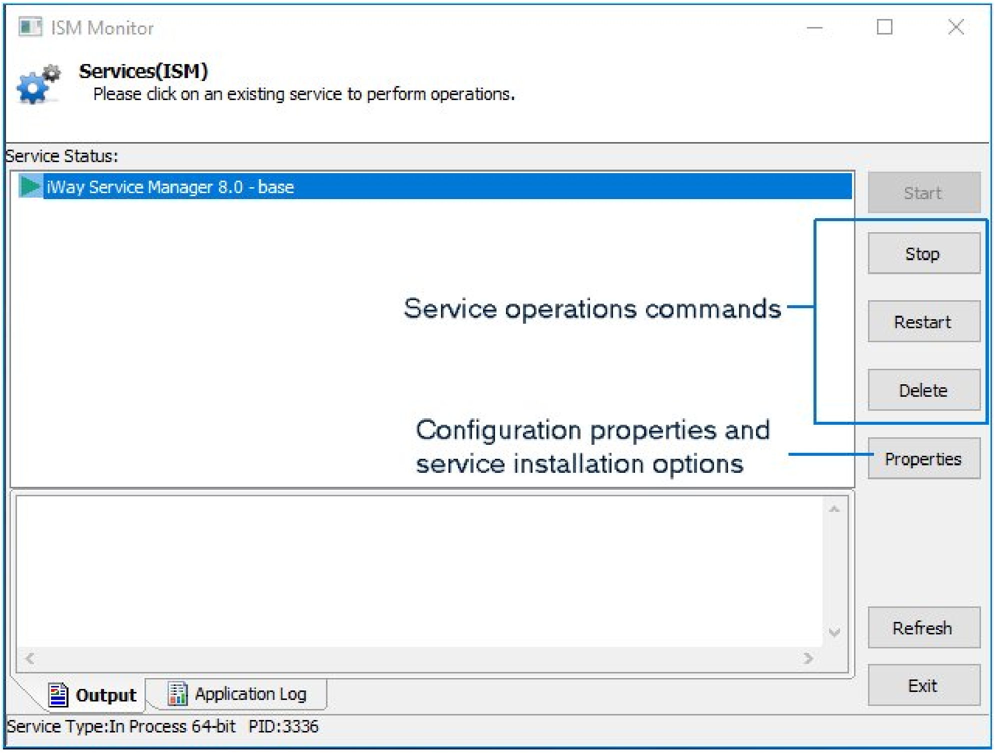
```
- <SOAP-ENV:Envelope>
  - <SOAP-ENV:Body>
    - <ivpResponse cid="AA413F44BE6EC5167E41A06DA22D289B">
        <CurrentTime>2009-12-14T19:14:11Z</CurrentTime>
        <Version>iWay 6.0.1 Service Manager (ga.22561)</Version>
      </ivpResponse>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
```

Using the iWay Service Manager System Tray Application on Windows

When iWay Service Manager (iSM) is installed in a Windows environment, an optional icon is added to the System Tray. This icon allows an administrator to control (for example, start, stop, restart, and delete) the iSM configurations that have been defined in the iSM Administration Console.

The iSM System Tray (iwsystray.exe) application also provides a convenient way to access and control iSM services in a Windows environment. An iSM icon that is associated with the application is available from the System Tray.

You can right-click or double-click the iSM icon to manage iSM configurations and services, as shown in the following image.



ISM Monitor

Services(ISM)

Please click on an existing service to perform operations.

Service Status:

iWay Service Manager 8.0 - base

Service Type:In Process 64-bit PID:3336

Service operations commands

Configuration properties and service installation options

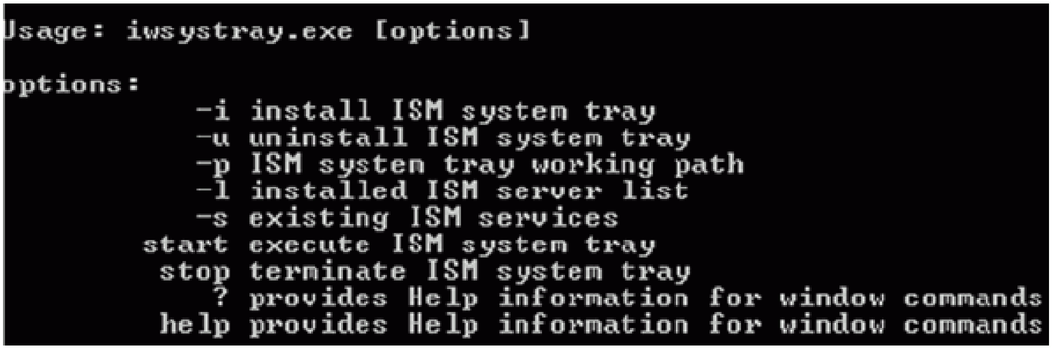
The iSM System Tray allows you to:

□ Control multiple iSM instances and managed configurations.

□ Monitor iSM configurations and application logs.

Help for command line options is available by executing the *iwsystray.exe* file, which is located in the following directory:

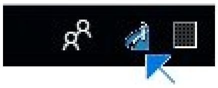
```
<iWayHome>\bin\iwsystray.exe
```



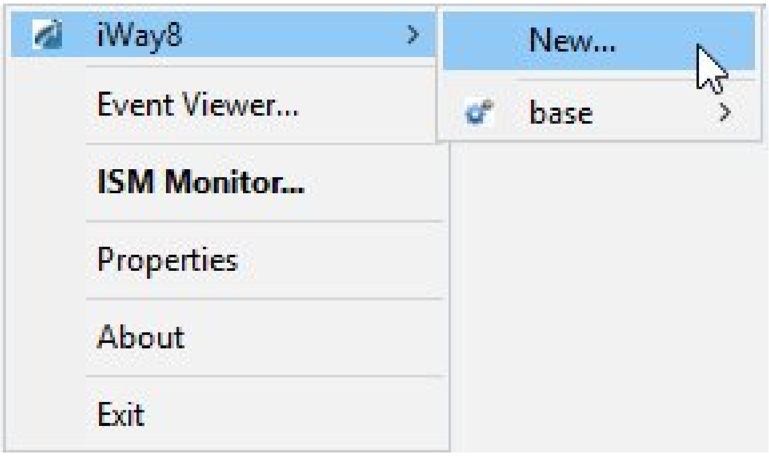
```
Usage: iwsystray.exe [options]

Options:
           -i install ISM system tray
           -u uninstall ISM system tray
           -p ISM systen tray working path
           -l installed ISM server list
           -s existing ISM services
        start execute ISM system tray
         stop terminate ISM system tray
            ? provides Help information for window commands
         help provides Help information for window commands
```

Configuring a New Service

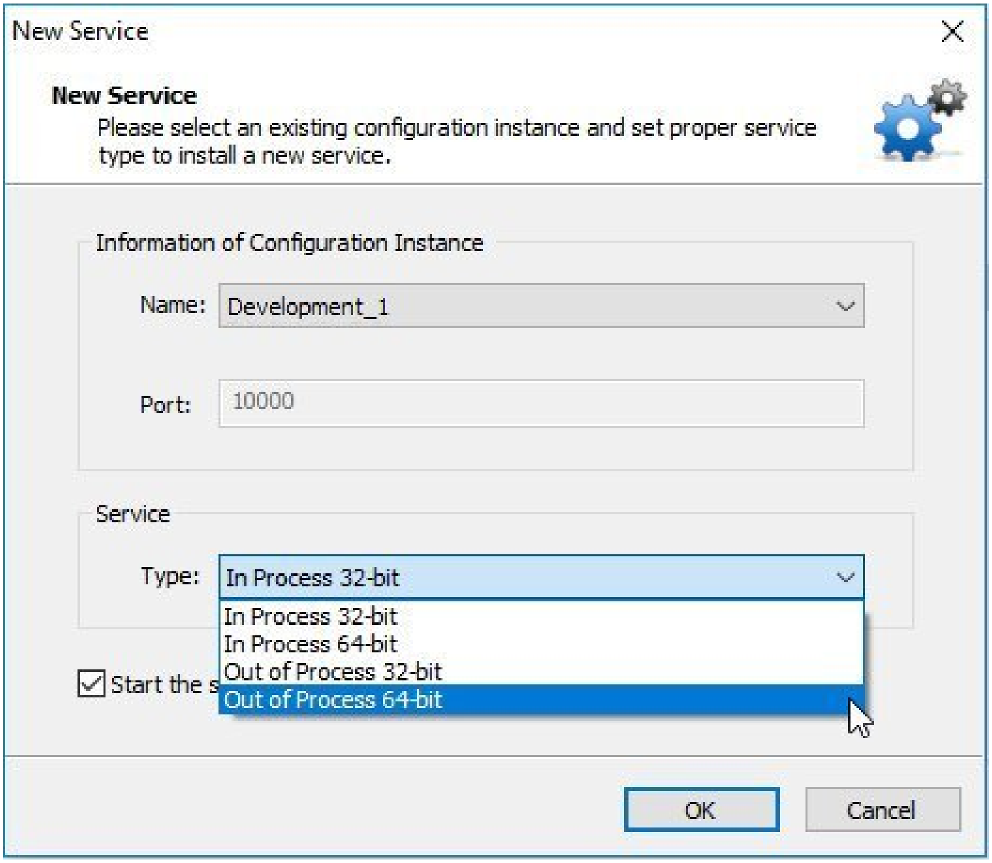
Right-click the iWay icon located in the Windows System Tray, as shown in the following image.



From the context menu that displays, select *iWay8* and then click *New*, as shown in the following image.



The New Service dialog box opens, as shown in the following image.



New Service

New Service

Please select an existing configuration instance and set proper service type to install a new service.

Information of Configuration Instance

Name: Development_1

Port: 10000

Service

Type: In Process 32-bit

☑ Start the s

Select the iSM configuration for which you want to create a service from the Name drop-down list (for example, Development_1). The defined port number for the iSM configuration is shown in the Port field.

The Type drop-down list provides the following service types you can assign for the selected iSM configuration:

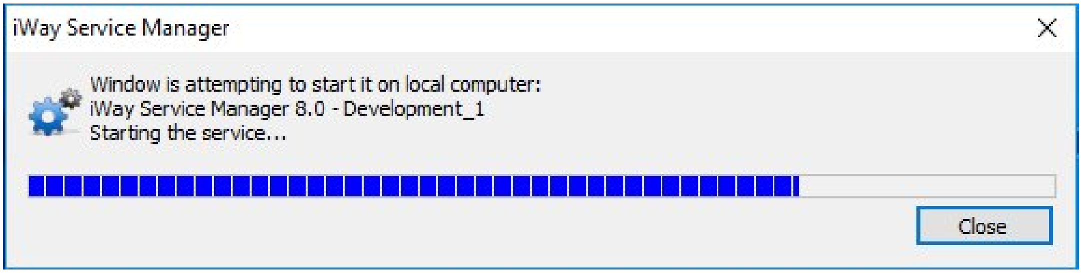
- ❑ In Process 32-bit
- ❑ In Process 64-bit
- ❑ Out of Process 32-bit
- ❑ Out of Process 64-bit

The 32 or 64-bit processing is dependent on the version of Java that was installed in the current Windows system that is being used. If the Java version is only 32-bit, then you will not have the ability to select the 64-bit service types.

After you make your selections, click *OK* to continue.

A set of status/progress dialogs appear in the following order:

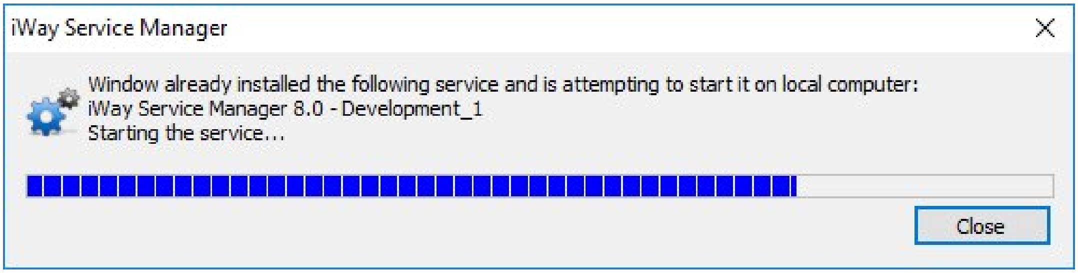
1. The *install* dialog, as shown in the following image.



iWay Service Manager

Window is attempting to start it on local computer:
iWay Service Manager 8.0 - Development_1
Starting the service...

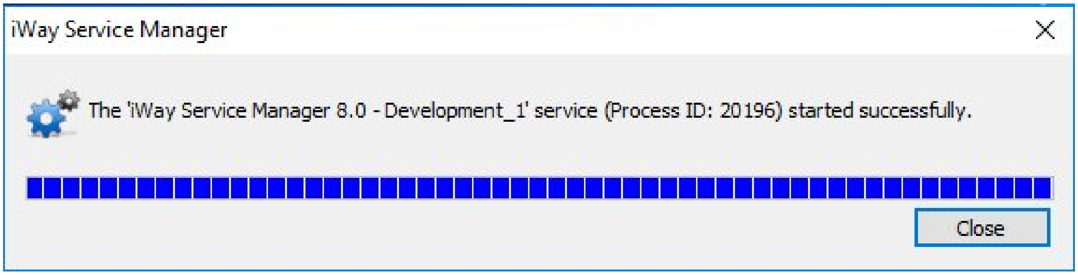
2. The *startup* dialog, as shown in the following image.



iWay Service Manager

Window already installed the following service and is attempting to start it on local computer:
iWay Service Manager 8.0 - Development_1
Starting the service...

3. The *final status* dialog, as shown in the following image.



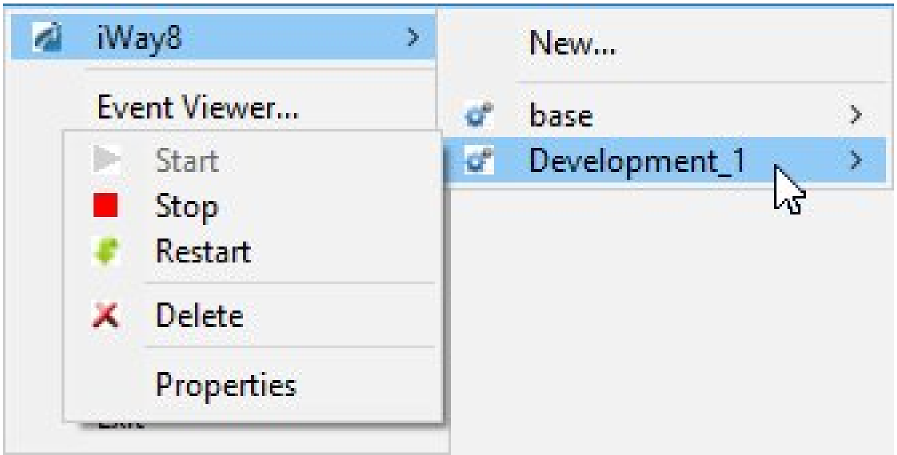
iWay Service Manager

The 'iWay Service Manager 8.0 - Development_1' service (Process ID: 20196) started successfully.

After the new service has installed and started successfully, click *Close* in the *final status* dialog.

Right-click the iWay icon located in the Windows System Tray.

From the context menu that displays, select *iWay8*, and then the name of the iSM configuration (for example, Development_1) for which you created the new service, as shown in the following image.



You, as an administrator, can now start, stop, restart or delete any defined iSM configuration in a Windows environment.

Note: When you use the iWay icon located in the Windows System Tray to create a new service for an iSM configuration, the entry that is created will be defined as an automatic startup service. This means that every time that the Windows system is started, this defined service will automatically start. In a Production environment this is probably what is required.

However, In a Development environment, you may want the ability to start and stop the service as needed. It is recommended for iSM services be started in this manner to maintain control. Therefore, after a service is defined, it is recommended that the administrator navigate to the Windows Control Panel (specifically Administrative Tools > Computer Management > Services and Applications > Services) and change the property of a defined iSM service from *automatic* startup to *manual* startup. This enables you to start and stop iSM services quickly for any defined configuration using the iWay icon located in the Windows System Tray.

Chapter 11

Using the Telnet Management Connection

This section provides an overview of the Telnet Management Connection and describes how to use this facility to remotely manage servers, whether running as a shell process or a service.

For more information, see the *iWay Service Manager Extensions User's Guide*.

For a list of commands that can be used on the Telnet Listener, see the *iWay Service Manager Command Reference Guide*.

In this chapter:

Telnet Management Connection Overview

The Telnet connection, meeting appropriate RFCs, makes a sub-selection of the standard shell management console available to the connected users.

Before the Telnet Management Connection can be used, the *telnet* extension must be installed and configured on the iSMO server. The Telnet user can use any standard Telnet client program to interact with the server.

Security Considerations

The Telnet Management Connection provides the following mechanisms to handle security considerations:

1. Iwtelnet must specifically configured.
2. An optional whitelist that accepts client connections only from authorized IP addresses.
3. A user ID and password with user and administrator levels. The password and roles are validated through the authentication realm mechanism, discussed below.
4. An optional SSL requirement permitting access only from SSL-enabled clients.
5. A timeout setting to prevent the impersonation of absent logged on users.

Configuring the Telnet Listener

The Telnet listener can be configured much like any other listener in iWay Service Manager. Once the listener is configured, it is then assembled and built into a channel. No other components are necessary for this channel. The channel must be active for the Telnet Management Connection to be used.

To prevent conflicts and accidental connections, iWay strongly suggests specifying a unique port number (instead of the standard default port number 23) during the configuration process.

Procedure: How to Configure the Telnet Listener

To configure the Telnet listener:

1. Ensure that iWay Service Manager is running.

   On Windows, you can start iWay Service Manager by clicking *Start*, selecting *Programs*, *iWay 8.0 Service Manager*, and then *Start Service Manager* for the configuration you are currently using.

   For more information on starting and stopping iWay Service Manager, see *Operations and Monitoring* on page 389.

2. Open a browser window and point to the following URL:

   ```
   http://host:port/ism
   ```

   where:

   host

   Is the host machine on which iWay Service Manager is installed.

   port

   Is the port on which iWay Service Manager is listening. The default port is 9999.

   On Windows, alternatively, you can click *Start*, select *Programs*, *iWay 8.0 Service Manager*, and then click *Console*.

   A login dialog box opens.

3. Type a user name and password for the configuration you are using, and click *OK*.

   The iWay Service Manager Administration Console opens.

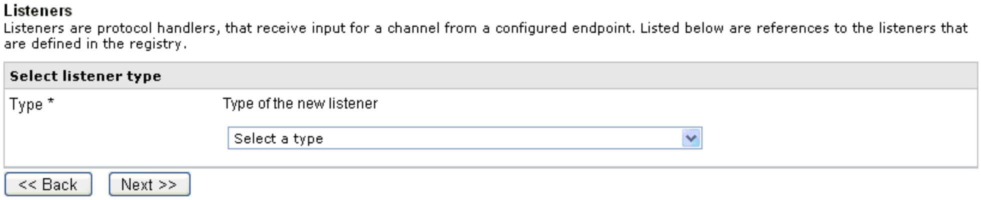
4. Click *Registry* in the top pane, and then click *Listeners* in the left pane.

   The Listeners pane opens.

   The table that is provided lists all the previously configured listeners and a brief description for each.

5. Click *Add*.

   The Select listener type pane opens.



Listeners

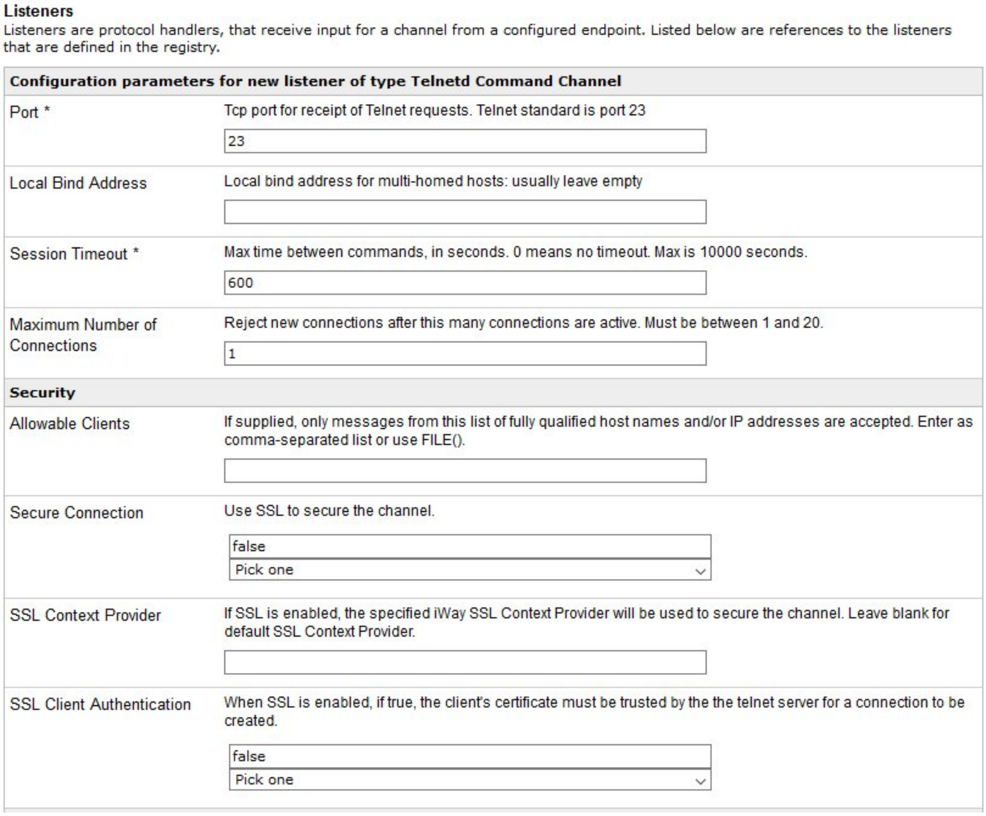
Listeners are protocol handlers, that receive input for a channel from a configured endpoint. Listed below are references to the listeners that are defined in the registry.

Select listener type

Type *	Type of the new listener Select a type

6. Select *Telnetd Command Channel* from the Type drop-down list and click *Next*.

   The Configuration parameters for the Telnetd listener pane opens.



Listeners

Listeners are protocol handlers, that receive input for a channel from a configured endpoint. Listed below are references to the listeners that are defined in the registry.

Configuration parameters for new listener of type Telnetd Command Channel	
Port *	Tcp port for receipt of Telnet requests. Telnet standard is port 23 23
Local Bind Address	Local bind address for multi-homed hosts: usually leave empty
Session Timeout *	Max time between commands, in seconds. 0 means no timeout. Max is 10000 seconds. 600
Maximum Number of Connections	Reject new connections after this many connections are active. Must be between 1 and 20. 1
Security	
Allowable Clients	If supplied, only messages from this list of fully qualified host names and/or IP addresses are accepted. Enter as comma-separated list or use FILE().
Secure Connection	Use SSL to secure the channel. false Pick one
SSL Context Provider	If SSL is enabled, the specified iWay SSL Context Provider will be used to secure the channel. Leave blank for default SSL Context Provider.
SSL Client Authentication	When SSL is enabled, if true, the client's certificate must be trusted by the the telnet server for a connection to be created. false Pick one

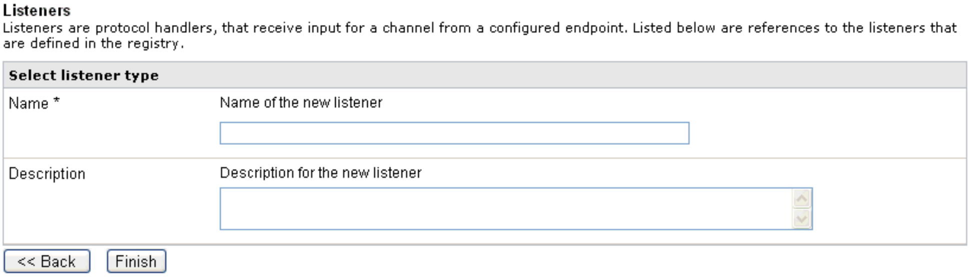
Note: The parameters prefixed with a (*) in the listener configuration pane are required.

7. Provide the appropriate values for the Telnetd listener parameters.

   For more information, see *Telnet Listener Configuration Parameters* on page 466.

8. Click *Next*.

   You are returned to the Select listener type pane.



Listeners

Listeners are protocol handlers, that receive input for a channel from a configured endpoint. Listed below are references to the listeners that are defined in the registry.

Select listener type

Name *	Name of the new listener
Description	Description for the new listener

9. Enter a name for the TCP listener and description (optional).
10. Click *Finish*.

    You can now use this listener as part of your channel configuration where the business logic will be applied to the received messages.

Reference: Telnet Listener Configuration Parameters

The following table lists and describes parameters for the Telnet listener.

Parameter	**Description**
Port	Telnet port for receipt of Telnet requests. **Note:** To prevent conflicts and accidental connections, iWay strongly suggests specifying a unique port number (instead of the standard default port number 23) during the configuration process.
Local bind address	The local bind address for multi-homed hosts. This parameter value is usually not specified.
Session Timeout	The maximum time between commands, in seconds. 0 means no timeout.The maximum is 10,000 seconds.
Maximum Number of Connections	Reject new connections after this many connections are active. Must be between 1 and 20.

Security

Parameter	Description
Allowable Clients	If supplied, only messages from this list of fully qualified host names and/or IP addresses are accepted. Accepts comma-separated list or use the FILE() function.
Allowable Access Attempts	Use SSL to secure the channel. By default, this parameter is set to *false*.
SSL Context Provider	Named iWay Security provider for SSL Context.
SSL Client Authentication	If set to True, the authentication will be required from the client.

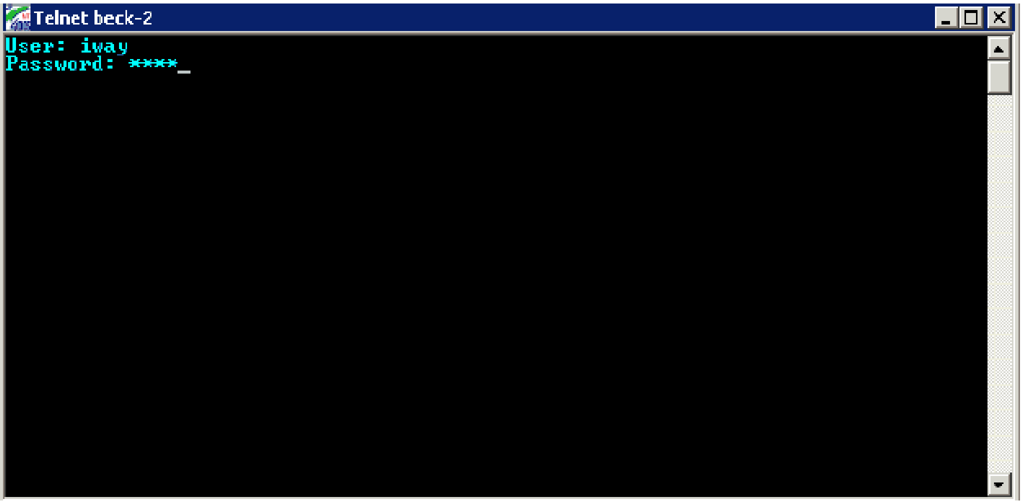
Login Security

Parameter	Description
Security	
Authentication Realm	Name of a configured authentication realm to validate logins. For full access to management commands, the user must be granted administrative privileges (admin role). If an authentication realm is not supplied, the logins will be delegated to the user database of the web console.
Allowable Access Attempts	The number of access attempts that will be allowed before invoking the Access Denied Flow.
Access Denied Flow	The optional iSM process flow to call when user fails to login within the Allowable Access Attempts.

Using the Telnet Management Connection

After you have installed the *Telnet* extension and configured a telnet listener, you can use the Telnet Management Connection. In this section, Microsoft's default Telnet client is used as an example.

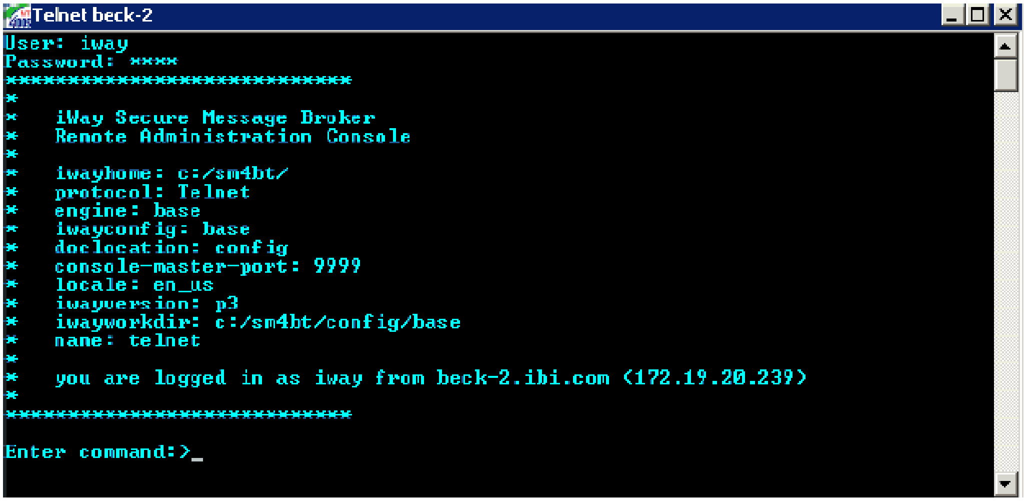
Once you start the Telnet client, the following Telnet logon screen is displayed.



```
User: iway
Password: ****_
```

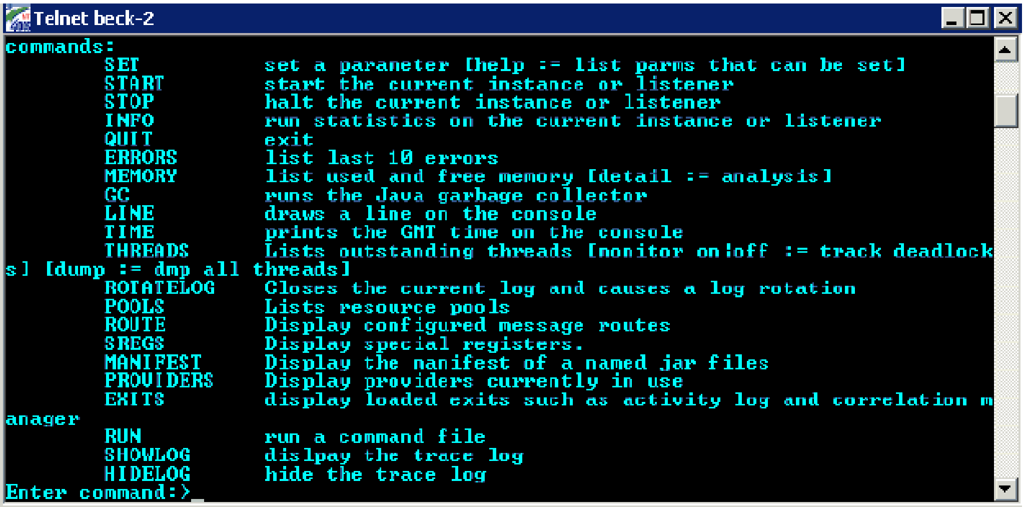
Provided that the connection meets the selected security criteria (whitelist SSL) you are prompted for a user ID and password. These must be configured in the iWay Management Console, and may have administrative capabilities or not. Lack of administrative capability means that commands that reconfigure the server, such as *start*, *stop* and *reinit* are not available.

Once the logon is accepted, you are presented with a standard information screen.



```
User: iway
Password: ****
****************************
*
*   iWay Secure Message Broker
*   Remote Administration Console
*
*   iwayhome: c:/sm4bt/
*   protocol: Telnet
*   engine: base
*   iwayconfig: base
*   doclocation: config
*   console-master-port: 9999
*   locale: en_us
*   iwayversion: p3
*   iwayworkdir: c:/sm4bt/config/base
*   name: telnet
*
*   you are logged in as iway from beck-2.ibi.com (172.19.20.239)
*
****************************

Enter command:>_
```

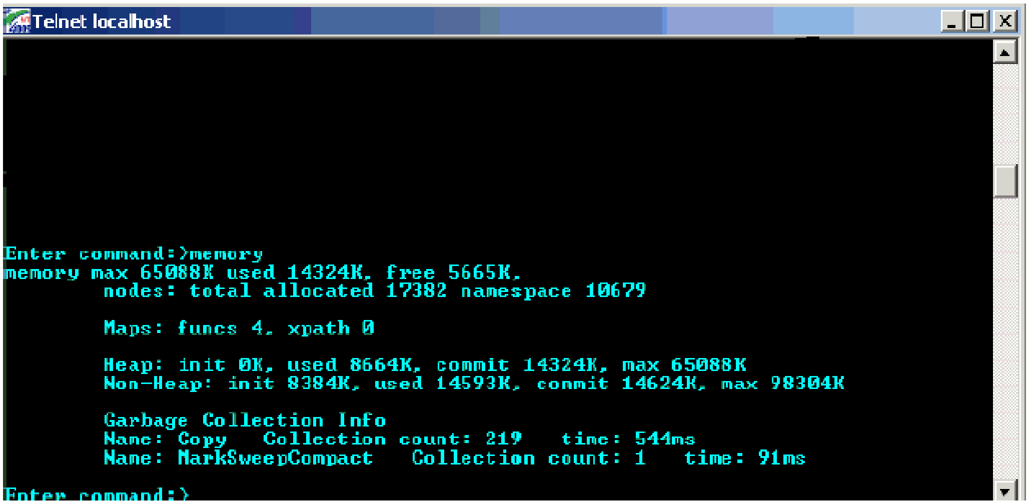
At the command line, you can use any authorized command. The *help* command lists these commands.



```
commands:
        SET             set a parameter [help := list parms that can be set]
        START           start the current instance or listener
        STOP            halt the current instance or listener
        INFO            run statistics on the current instance or listener
        QUIT            exit
        ERRORS          list last 10 errors
        MEMORY          list used and free memory [detail := analysis]
        GC              runs the Java garbage collector
        LINE            draws a line on the console
        TIME            prints the GMT time on the console
        THREADS         Lists outstanding threads [monitor on|off := track deadlock
s] [dump := dmp all threads]
        ROTATELOG       Closes the current log and causes a log rotation
        POOLS           Lists resource pools
        ROUTE           Display configured message routes
        SREGS           Display special registers.
        MANIFEST        Display the manifest of a named jar files
        PROVIDERS       Display providers currently in use
        EXITS           display loaded exits such as activity log and correlation m
anager
        RUN             run a command file
        SHOWLOG         dislpay the trace log
        HIDELOG         hide the trace log
Enter command:>_
```

These are the same commands that can be issued from the standard shell console, plus the *showlog* and *hidelog* commands to enable or disable tracing for this telnet session.

For example, if you enter the *memory* command, the following screen is displayed.



```
Enter command:>memory
memory max 65088K used 14324K, free 5665K.
        nodes: total allocated 17382 namespace 10679

        Maps: funcs 4, xpath 0

        Heap: init 0K, used 8664K, commit 14324K, max 65088K
        Non-Heap: init 8384K, used 14593K, commit 14624K, max 98304K

        Garbage Collection Info
        Name: Copy    Collection count: 219    time: 544ms
        Name: MarkSweepCompact    Collection count: 1    time: 91ms

Enter command:>
```

Chapter 12

Diagnostics, Tracing, and Logging

This section describes how to use the iWay Service Manager Administration Console to perform diagnostic functions. Information on how to run iWay Service Manager (iSM) in a command shell to enter commands that can assist you during troubleshooting is provided. It also describes how to use the Windows Event Viewer for troubleshooting purposes.

The following topics include instructions on how to configure logging and debugging properties to view the resulting log files in the console. Instructions on how to enable the tracing features of iSM are also provided, in addition to information about the iSM test tools that assist in debugging.

In this chapter:

Running in a Command Shell

You can run iWay Service Manager (iSM) in a command shell to debug and troubleshoot any errors that may occur.

For more information on starting and stopping iSM in a command shell, see *Operations and Monitoring* on page 389.

On Windows, the *iwsrv* command starts iSM in a command window for debugging purposes. For reference, the following topic includes the full syntax of the *iwsrv* command.

Syntax: How to Start iWay Service Manager in a Command Window (Windows)

Navigate to the iWay home bin directory. For example, on Windows, if iWay is installed in C:\Program Files\iWay8, go to

```
C:\Program Files\iWay8\bin
```

The syntax for the *iwsrv* command, which starts iWay Service Manager in a command window on Windows is:

```
iwsrv [configuration] [-s service] [-l launch] [options]
```

where:

`configuration`

Is the name of the server configuration that is loaded for this instance. The default value is base.

`service`

Is the name of the service that is executed. Valid values are:

start: Starts the server configuration (default).

stop: Stops the server configuration.

install: Installs the server configuration.

remove: Removes the server configuration.

query: Queries the server configuration.

`launch`

Specifies the launch method. Valid methods are:

java: Loads Java in a separate process and uses the JVM options, NT dependencies, and other preferences found within the iSM configuration that are configured through the console. For example: `iwsrv.exe base -s start -l java`

script file: Specifies a script file that defines the run-time preferences. This script file must be located in the iWay Service Manager installation directory. For example: `iwsrv.exe base -s start -l iWay8.cmd`

Both of the above uses of -l will force the service to load Java in a separate process. When the service is stopped, both iwsrv.exe and java.exe are terminated.

options

Specifies tracing or server back-up information. Valid values include:

-b: Indicates that Service Manager is a back-up server, for example:

```
iwsrv.exe base -s start -b
```

-c: Turns tracing on. In this mode, you can display useful error messages on the console. For example, you can display a message that says the Java Runtime Environment (JRE) is not properly installed. For example:

```
iwsrv.exe base -s start -c
```

-d: Limits tracing to debug only, for example:

```
iwsrv.exe base -s start -d
```

-f: [PATH] filters the system path when invoking JAVA. [RESTART] suppresses the JVM fault restart capability.

-h: iWay8 home directory.

-t: The amount of time (in seconds) to process service shutdown.

Example: Starting a Server Configuration With Traces Enabled

The following command starts a server configuration named *test* and sends traces to the command window as print lines:

```
iwsrv test -c
```

Diagnostic Commands

When iWay Service Manager (iSM) is running in a command shell, you can control it by typing commands in addition to using the console. These commands are designed to assist you in resolving issues.

Commands enable you to examine registers and settings, change key settings, run specific process flows, and manage the system in general. Commands can be entered in any command shell window, or in a remote command shell (for example, Telnet and SSH) created through the Activity Facility of the iSM Administration Console.

To view a full list of available commands, type *Help* at the Command Prompt on Windows after using the *iwsrv* command to start iSM.

For more information on all of the available iSM commands, see the *iWay Service Manager Command Reference Guide*.

Troubleshooting on Windows

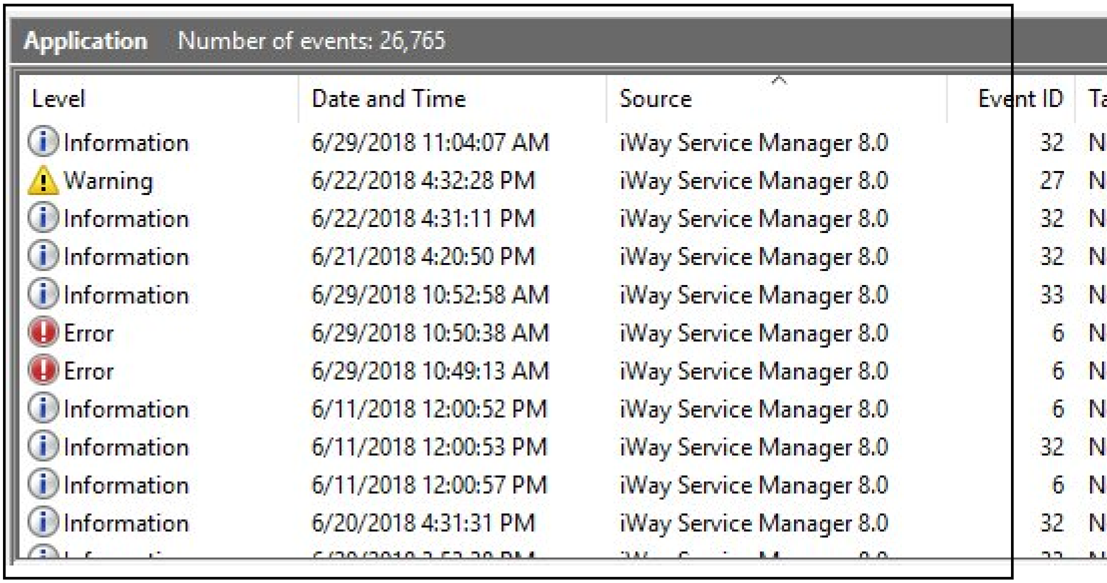
Information, warning, and error messages are logged in the Windows Event Log system. When a problem occurs, the Windows Event Log is the first place to look for information.

Procedure: How to Display Messages in the Windows Event Viewer

To display messages in the Windows Event Viewer:

1. Access the Event Viewer from Administrative Tools, which can be found in the Windows Control Panel.
2. From the left pane of the Event Viewer, click *Application* to view iWay Service Manager entries.

   The following image shows entries in columns that indicate the type of message, the date and time, the source (for example, iWay Service Manager), the category, the event, and the user, when applicable.



Application Number of events: 26,765

Level	Date and Time	Source	Event ID	Ta
Information	6/29/2018 11:04:07 AM	iWay Service Manager 8.0	32	N
Warning	6/22/2018 4:32:28 PM	iWay Service Manager 8.0	27	N
Information	6/22/2018 4:31:11 PM	iWay Service Manager 8.0	32	N
Information	6/21/2018 4:20:50 PM	iWay Service Manager 8.0	32	N
Information	6/29/2018 10:52:58 AM	iWay Service Manager 8.0	33	N
Error	6/29/2018 10:50:38 AM	iWay Service Manager 8.0	6	N
Error	6/29/2018 10:49:13 AM	iWay Service Manager 8.0	6	N
Information	6/11/2018 12:00:52 PM	iWay Service Manager 8.0	6	N
Information	6/11/2018 12:00:53 PM	iWay Service Manager 8.0	32	N
Information	6/11/2018 12:00:57 PM	iWay Service Manager 8.0	6	N
Information	6/20/2018 4:31:31 PM	iWay Service Manager 8.0	32	N

3. To view its contents, double-click an entry.

   If you are having difficulty starting a service for iWay Service Manager, which cannot be resolved using information from the Event Viewer, start the service in console mode.

Performing Diagnostic Functions

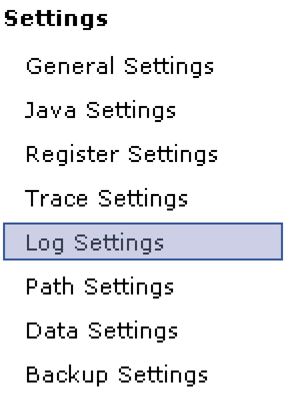
The iWay Service Manager Administration Console enables you to configure diagnostic properties for logging and tracing. After logging and tracing properties are enabled, you can view the resulting log files in the console.

Log Settings

The Trace Log is used to record the diagnostic information that is generated by the run-time components of iWay Service Manager. The Transaction Log is used to maintain a record of every document received and processed by iWay Service Manager. The following procedure describes how to configure log settings that are defined in the base configuration of iWay Service Manager.

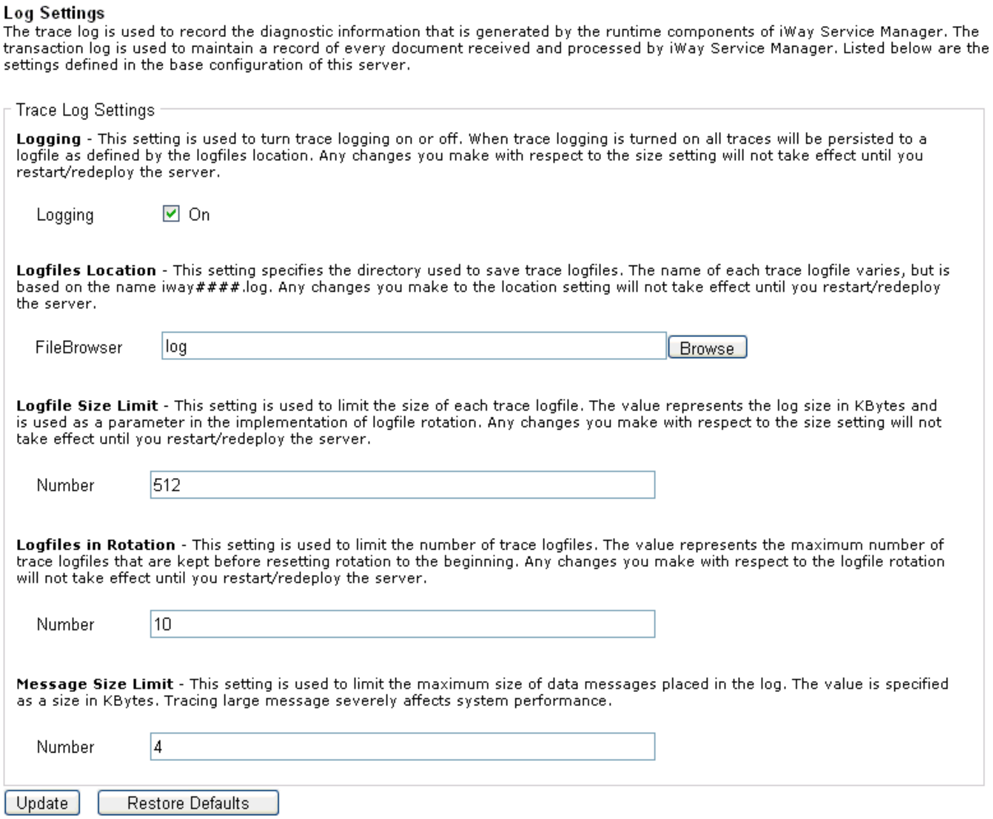
Procedure: How to Configure Log Settings

To configure log settings:



1. In the left console pane of the Server menu, select *Log Setting*s.

The Log Settings pane displays, as shown in the following image.



Log Settings

The trace log is used to record the diagnostic information that is generated by the runtime components of iWay Service Manager. The transaction log is used to maintain a record of every document received and processed by iWay Service Manager. Listed below are the settings defined in the base configuration of this server.

Trace Log Settings

Logging - This setting is used to turn trace logging on or off. When trace logging is turned on all traces will be persisted to a logfile as defined by the logfiles location. Any changes you make with respect to the size setting will not take effect until you restart/redeploy the server.

Logging ☑ On

Logfiles Location - This setting specifies the directory used to save trace logfiles. The name of each trace logfile varies, but is based on the name iway####.log. Any changes you make to the location setting will not take effect until you restart/redeploy the server.

FileBrowser log

Logfile Size Limit - This setting is used to limit the size of each trace logfile. The value represents the log size in KBytes and is used as a parameter in the implementation of logfile rotation. Any changes you make with respect to the size setting will not take effect until you restart/redeploy the server.

Number 512

Logfiles in Rotation - This setting is used to limit the number of trace logfiles. The value represents the maximum number of trace logfiles that are kept before resetting rotation to the beginning. Any changes you make with respect to the logfile rotation will not take effect until you restart/redeploy the server.

Number 10

Message Size Limit - This setting is used to limit the maximum size of data messages placed in the log. The value is specified as a size in KBytes. Tracing large message severely affects system performance.

Number 4

2. Change the default values.

   For more information, see *Log Setting Properties* on page 476.

3. Click *Update*.

4. For your changes to take effect, restart iWay Service Manager.

Reference: Log Setting Properties

The following table lists and describes the log setting properties.

Property	Description
Trace Log Settings	

Property	Description
Logging	Turns logging on or off. Required if you want to log to a file, use a diagnostic activity log, or view the log online.
Logfiles Location (Directory field)	Directory where the trace log root resides. To create the directory if it does not exist, select the check box.
Logfile Size Limit (Number field)	Maximum allowed for each file size in kilobytes (used for log rotation). iWay recommends a minimum of one megabyte.
Logfiles in Rotation (Number field)	Maximum number of files to keep (used for log rotation).
Message Size Limit (Number field)	Maximum size of the data message in a log file measured in kilobytes. Large trace messages affect system performance.

Trace Settings

Tracing is key to diagnosing problems and thus to application reliability. iWay Service Manager provides a full complement of tracing services, oriented to diagnostic analysis of the running system. Tracing provides a step-by-step explanation of the internal activity of the server.

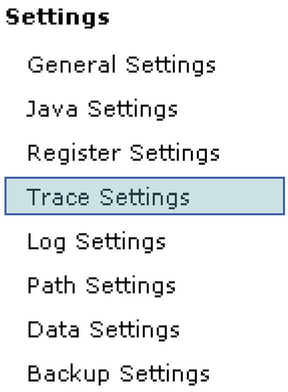
It is important to note that tracing can affect system performance. The iWay Service Manager Administration Console enables you to select the levels of traces that you want to generate. Unless you are diagnosing a problem, you should limit tracing to error-level only.

A separate category called JLINK debug masks trace messages originating in the iWay JDBC driver that is used to access the main data server. You can specify actual tracing levels for all instances of the driver in the driver settings of the Data Server Properties configuration window. For more information, see *How to Activate JLINK Tracing* on page 481.

The following procedure describes how to control the amount of detail that is produced by the diagnostic components embedded within iSM. Traces produced during run time are displayed or logged based on settings in the run-time environment.

Procedure: How to Select Trace Levels

To select trace levels:



Settings

General Settings

Java Settings

Register Settings

Trace Settings

Log Settings

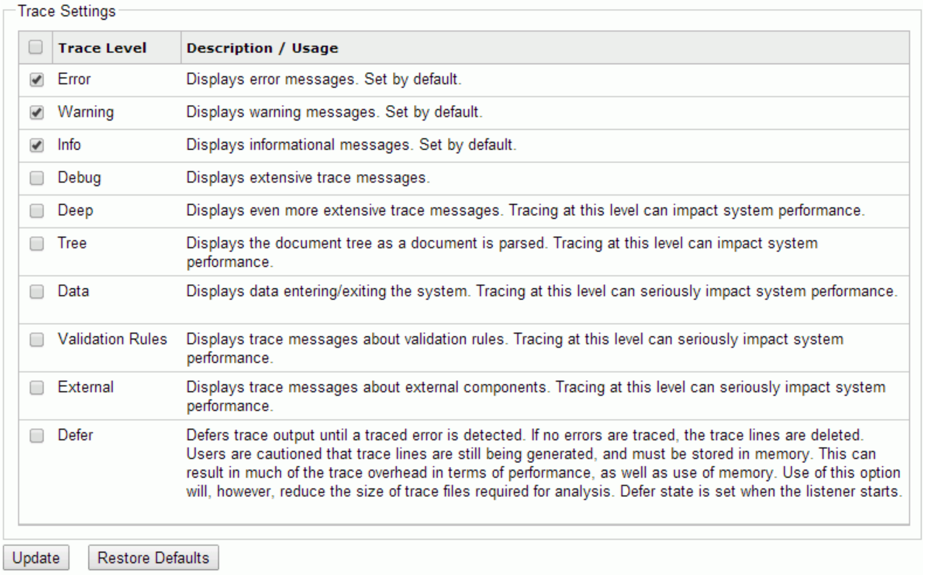
Path Settings

Data Settings

Backup Settings

1. In the left console pane of the Server menu, select *Trace Settings*.

   The Trace Settings pane displays, as shown in the following image.



Trace Settings

☐	Trace Level	Description / Usage
☑	Error	Displays error messages. Set by default.
☑	Warning	Displays warning messages. Set by default.
☑	Info	Displays informational messages. Set by default.
☐	Debug	Displays extensive trace messages.
☐	Deep	Displays even more extensive trace messages. Tracing at this level can impact system performance.
☐	Tree	Displays the document tree as a document is parsed. Tracing at this level can impact system performance.
☐	Data	Displays data entering/exiting the system. Tracing at this level can seriously impact system performance.
☐	Validation Rules	Displays trace messages about validation rules. Tracing at this level can seriously impact system performance.
☐	External	Displays trace messages about external components. Tracing at this level can seriously impact system performance.
☐	Defer	Defers trace output until a traced error is detected. If no errors are traced, the trace lines are deleted. Users are cautioned that trace lines are still being generated, and must be stored in memory. This can result in much of the trace overhead in terms of performance, as well as use of memory. Use of this option will, however, reduce the size of trace files required for analysis. Defer state is set when the listener starts.

2. If other than the default trace levels (Info and Error) are required, select the desired trace level check box.

   For more information, see *Trace Setting Properties* on page 479.

3. Click *Update*.

Reference: Trace Setting Properties

The following table lists and describes the trace setting properties.

Trace Level	Description
Error	Displays error messages. This trace level is set by default.
Warning	Displays warning messages. This trace level is set by default.
Info	Displays informational messages. This trace level is set by default.
Debug	Reports data that is helpful for debugging situations. Shows logic that tracks the path of a document.
Deep	Used for detailed logic tracing. Stack traces are reported by the system in deep debug level. Use only if instructed to do so by iWay Support. **Caution:** Tracing at this level can impact system performance.
Tree	Displays the document as it enters and leaves the system in XML form. This is a level at which intermediate processing as a document evolves is done. **Caution:** Tracing at this level can impact system performance.
Data	Displays the incoming and outgoing documents as they pass to and from the protocol channel. **Caution:** Tracing at this level can impact system performance.
Rules	Displays trace messages about rules. **Caution:** Tracing at this level can impact system performance.
External	Displays trace messages about external components. **Caution:** Tracing at this level can impact system performance.

Trace Level	Description
Defer	Defers trace output until a traced error is detected. If no errors are traced, the trace lines are deleted. Use of this option will, however, reduce the size of trace files required for analysis. Defer state is set when the listener starts. **Caution:** Trace lines are still being generated, and must be stored in memory. This can result in much of the trace overhead in terms of performance, as well as use of memory.

Trace levels also can be set on and off by using the *set* command. For example:

```
set tree on
```

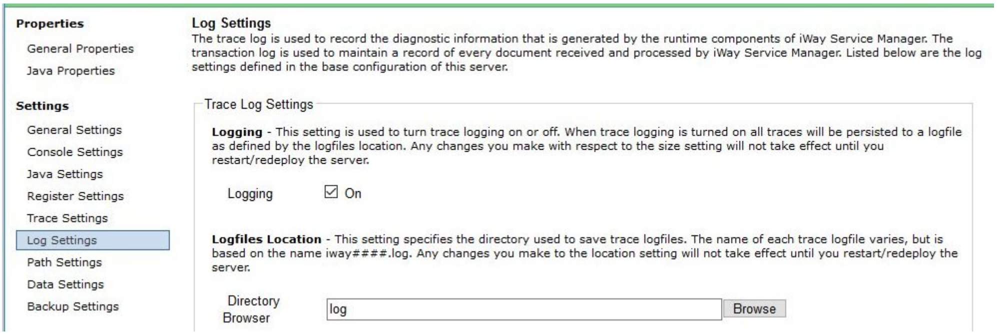
Procedure: How to Log Traces to a File

If tracing is turned on without logging, the tracing information appears only in the debug window and is not saved to a file.

To log traces to a file:

1. In the left console pane of the Server menu, select *Log Settings*.

   The Log Settings pane opens.

   Log Settings

   The trace log is used to record the diagnostic information that is generated by the runtime components of iWay Service Manager. The transaction log is used to maintain a record of every document received and processed by iWay Service Manager. Listed below are the log settings defined in the base configuration of this server.

   Trace Log Settings

   Logging - This setting is used to turn trace logging on or off. When trace logging is turned on all traces will be persisted to a logfile as defined by the logfiles location. Any changes you make with respect to the size setting will not take effect until you restart/redeploy the server.

   Logging ☑ On

   Logfiles Location - This setting specifies the directory used to save trace logfiles. The name of each trace logfile varies, but is based on the name iway####.log. Any changes you make to the location setting will not take effect until you restart/redeploy the server.

   Directory Browser: log

2. In the Logfiles Location section, specify the path to the directory used to save log files.
3. Click *Update*.
4. For your changes to take effect, restart iWay Service Manager.

Traces are available in several levels and controlled independently. For more information, see *Trace Setting Properties* on page 479.

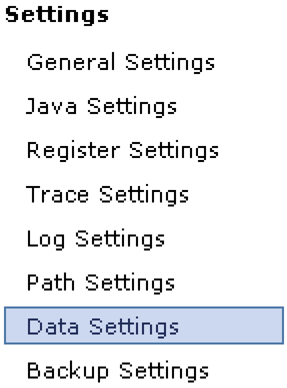
Note: Trace settings for managed configurations must be set for each configuration independently.

All levels can be masked, so that the log contains only brief informational and error messages.

Unlike most design time settings, changing trace levels takes immediate effect in the run-time system. Changing the log file location does not take effect until iWay Service Manager is restarted.

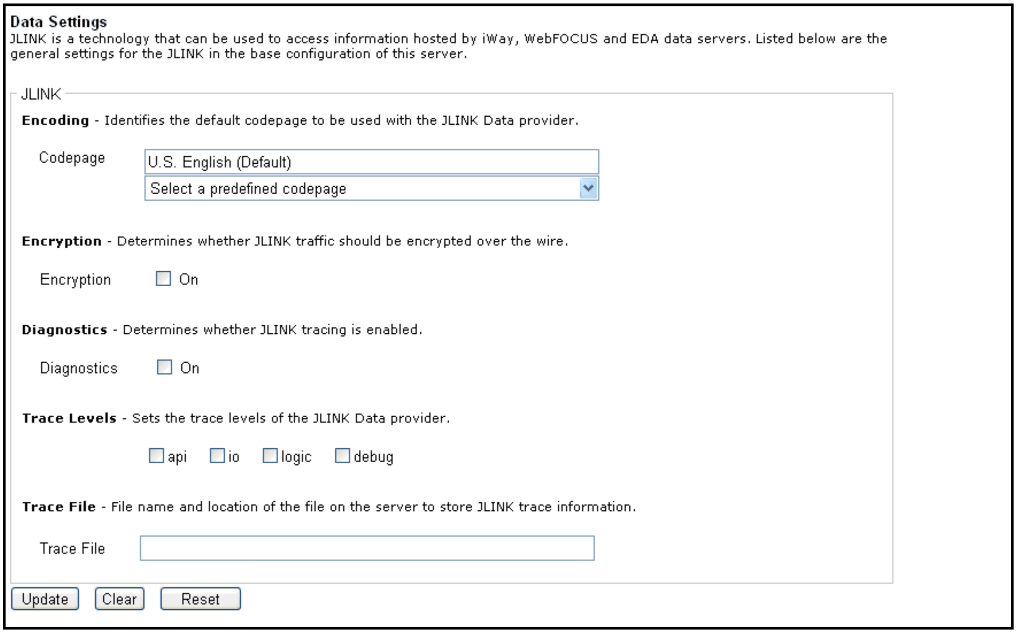
Procedure: How to Activate JLINK Tracing

To activate JLINK tracing in iWay Service Manager:



1. In the left console pane of the Server menu, select *Data Settings*.

The Data Settings pane opens, as shown in the following image.



Data Settings

JLINK is a technology that can be used to access information hosted by iWay, WebFOCUS and EDA data servers. Listed below are the general settings for the JLINK in the base configuration of this server.

JLINK

Encoding - Identifies the default codepage to be used with the JLINK Data provider.

Codepage: U.S. English (Default)
Select a predefined codepage

Encryption - Determines whether JLINK traffic should be encrypted over the wire.

Encryption □ On

Diagnostics - Determines whether JLINK tracing is enabled.

Diagnostics □ On

Trace Levels - Sets the trace levels of the JLINK Data provider.

□ api □ io □ logic □ debug

Trace File - File name and location of the file on the server to store JLINK trace information.

Trace File

a. Select the *Diagnostics* check box.

b. Specify trace levels for specific instances of the driver.

The trace levels are:

api. Provides entry/exit tracing as the application steps through JDBC calls.

io. Traces data in and out of the system.

logic. Traces the internal activity of the driver. This is equivalent to the Debug trace level of the server.

debug. Traces internal operations of the driver. This is equivalent to the Deep Debug trace level of the server.

c. Type the name of the trace file in the Trace File field.

The trace file specification enables you to route traces from the iWay JDBC driver to a specific file. If you do not specify a trace file, the traces (in most cases) appear in the standard server trace. Certain generalized services that use the iWay JDBC driver do not pass traces through the server. In these cases, specification of the external trace file enables the traces to be captured. You may be prompted to send this file to iWay Support as part of the problem resolution process.

2. Click *Update*.

Measurements and Statistics

The Measurements package allows you to analyze the behavior of iWay Service Manager.

The Statistics package provides the following functionality:

- Reports heap memory usage as an extension to the existing **memory** command.
- Searches for and detects deadlocked workers as part of the extended **threads** command.
- Reports CPU and user time expended by masters as part of the extended **stats** command. Usage statistics can also be sent to an external monitoring facility for more detailed analysis.

In some situations, the Measurements package can add significant overhead to the operation of iWay Service Manager. Therefore, do not use the Measurements package in a production environment unless that environment is undergoing analysis.

The information and formats described in this topic are release-dependent, and subject to change.

Show Memory

The **show memory** command displays the amount of memory in use at the time that the command is issued. When the Statistics package is in use, the standard display is augmented by an additional line that starts with the word Heap.

The Java Virtual Machine has a heap, which is the run-time data area from which all required memory is allocated. The heap is created at the startup of the Java Virtual Machine. Heap memory for objects is reclaimed by an automatic memory management system, which is known as a *garbage collector*. Although the garbage collector runs automatically, you can issue the **gc** command to force it to run for analytic purposes.

The heap memory display has four fields, as listed and described in the following table.

Field	Description
init	Represents the initial amount of memory (in bytes) that the Java Virtual Machine requests from the operating system for memory management during startup. The Java Virtual Machine may request additional memory from the operating system, and may also release memory to the system over time. The value of *init* may be undefined (0) on some platforms.
committed	Represents the amount of memory (in bytes) that is guaranteed to be available for use by the Java Virtual Machine. The amount of *committed* memory may change over time (it may increase or decrease). The Java Virtual Machine may release memory to the system, and the value of *committed* could be less than the value of *init*. The value of *committed* is always greater than or equal to the value of *used*.
max	Represents the maximum amount of memory (in bytes) that can be used for memory management. Its value may be undefined. If its value is defined, the maximum amount of memory may change over time. If *max* is defined, the amount of *used* and *committed* memory is always less than or equal to the value of*max*. A memory allocation may fail if it attempts to increase the *used* memory, such that the value of *used* is greater than the value of *committed*, even if the value of *used* is less than or equal to the value of *max* (for example, when the system is low on virtual memory).
used	Represents the amount of memory currently used.

The heap display is more accurate than the memory display issued without the Measurements package installed. The original (standard) information is displayed, in addition to the new heap information.

```
Enter command:>memory
STR00X35: memory used 8244K, free 591K,
   nodes: cache 1001 allocated 8607, reclaimed 116, destroyed 116
   namespace 0 namespace reclaim 0
   Heap: init=0K committed=8244K max=65088K used=7662K

Enter command:>
```

The key value is *used*, which indicates how much memory is currently allocated. As the value of *used* approaches the value of *max*, the garbage collector may start, and performance may be eroded.

Deadlocks

The deadlock detector finds cycles of threads that are in deadlock, waiting to acquire locks. Deadlocked threads are blocked, waiting to enter a synchronization block, or waiting to reenter a synchronization block after a wait call, in which each thread owns one lock while trying to obtain another lock already held by another thread.

A thread is deadlocked if it is part of a cycle in the relation *is waiting for lock owned by*. In the simplest case, thread A is blocked, waiting for a lock owned by thread B, and thread B is blocked, waiting for a lock owned by thread A.

This is an expensive operation. Use it only in cases in which you suspect that messages are locked up in the system.

To enable monitoring, use the command **thread monitor on**. To disable monitoring, use the command **thread monitor off**. While the monitor is enabled, entering the **threads** command displays information regarding deadlocks, such as the thread name or names, and the lock name or names. The thread names indicate the components that are deadlocked.

Statistics

When iWay Service Manager is running without the Measurements package, some statistics are generated with wall clock times. With the Measurements package, the CPU and user state times are also generated. On the summary page, all values for time are reported in seconds, with a precision of four places. It is possible to develop a report with a greater precision for time.

In some cases, wall clock times show useful information. These times provide a measure of performance for a single message as experienced by the sender. They do not provide any information regarding the throughput capacity of iWay Service Manager.

CPU and user times describe the actual execution time expended on messages. Implementation of these measurements depends on the platform and the Java Virtual Machine (JVM). In many cases, the CPU and user times are the same, as the JVM may not discriminate between the two. For those platforms on which the JVM does discriminate, expect the CPU time to be greater than the user time.

User time is the CPU time that the current thread has executed in user mode, that is, the time spent executing iWay Service Manager instructions.

CPU time is the sum of user time and system time. It includes the time spent setting up for JVM services such as locks, network operations, I/O operations, and other services.

```
Enter command:>stats
       In seconds
   name      count     low    high    mean variance  std.dev. ehr num/sec
mq1a      wall:  2  0.0470  0.1560  0.1015   0.0030    0.0545   -    9.85
          cpu :  2  0.0312  0.0625  0.0469   0.0002    0.0156   -
          user:  2  0.0312  0.0625  0.0469   0.0002    0.0156   -
```

The **stats** command displays a summary of statistics gathered up to that point. To reset the values to zero, use the **stats reset** command. iWay recommends that you do not rely on statistics until several messages have been handled to completion, as iWay Service Manager front-loads initialization. Once the system is in a steady state, reset the statistics to zero.

The numbers displayed on the summary page are approximate and are intended for general guidance only. Brief descriptions of the displayed fields are provided in the following table. A fuller understanding of the message processing distribution described here requires some knowledge of statistics and probability, as they apply to queuing.

Field	Description
count	The number of messages that have been handled, for which statistics have been gathered.
low	The lowest time recorded for the handling of a message.
high	The highest time recorded for the handling of a message.
mean	The numeric mean of the times recorded. This value is the sum of the times divided by the number of messages handled. This value is frequently called the average.
variance	The statistical variance of the times recorded. Variance is a measure of how numbers disburse around the mean.
std.dev.	The statistical standard deviation of the times recorded. Standard deviation is a measure of how numbers disburse around the mean.

Field	Description
ehr	The Ehrlang Density Coefficient, which provides evidence of the randomness of the time distribution. If there are too few values to compute the coefficient, a hyphen (-) is displayed. If the coefficient is sufficiently close to constant, the term *const* is displayed. This value is an approximation. A value of 1.0 indicates a Poisson distribution, which is the design point of iWay Service Manager. A very low value can indicate that the individual times recorded are skewed and therefore less usable for predicting behavior.
num/sec	The reciprocal of the mean, providing the number of messages handled per second. This value is displayed for the wall time. It is not a direct measure of the throughput capacity of iWay Service Manager.

The iWay Service Manager Administration Console also displays a summary of statistics.

Emitted Statistics Information

iWay Service Manager can emit statistics as each measurement is generated. Statistics records are included in a comma-delimited file of alphanumeric characters.

The following table describes the fields in the file.

Field	Format	Description
type	String	The value 1, which is the record type. Other record types may be added in a future release.
id	String	The generator (worker) ID.
tid	String	The transaction ID.
msglen	Integer	The message length (non-streaming). If the length cannot be determined, the value -1 is specified.

Field	Format	Description
complexity	Integer	A measure of the complexity of the message. The higher the number, the greater the complexity. This value is generally a measure of the number of nodes in the XML tree. A value of -1 means unknown. For most purposes, the number of digits that this integer has is a good value to analyze.
timestamp	Integer	The current time, in milliseconds. This value is the difference, measured in milliseconds, between the current time and midnight, January 1, 1970 UTC (Universal Time, Coordinated). The value is a timestamp for the record.
gregtime	String	The timestamp in GMT (Greenwich Mean Time). The format is: yyyy-mm-dd**T**hh:mm:ss:mmm**Z**
walltime	Float	The wall clock time expended, in milliseconds.
usertime	Float	The user time expended, in milliseconds.
cputime	Float	The CPU time expended, in milliseconds.
usedmem	Integer	The used memory, in kilobytes (K). See *Show Memory* on page 483.
committed	Integer	The committed memory, in kilobytes (K). See *Show Memory* on page 483.
$	String	The end-of-record indicator.

The following is a sample record in the file:

```
1, W.udp1.1, udp1-UDP-W.udp1.1_20050328191546135Z, -1,1735, 1112037346213,
2005-03-28-T19:15:46:135Z, 78.0, 15.625, 15.625,1200,800 $
```

To enable iWay Service Manager to emit these statistics, define the following to the JVM properties

```
-Dstaturl=host:port
```

where *host* and *port* are a UDP receiver.

You can specify the host and port of an iWay Service Manager UDP listener that has a process that is defined to handle incoming messages. Use the iWay Service Manager Administration Console to help define Java system properties.

The iwmeasure.jar extension provides a simple StatsGather agent that appends each record to a named file.

Do not run the statistics gathering component on a machine that is being measured. The process of receiving the statistics will be measured, creating a loop. You must use two configurations, preferably on separate machines.

Tips

- ❑ When you work with the complexity of a document (a number greater than -1 in the complexity field), a good guideline is to use the number of digits in the field. For example, a message that consists of 172 nodes would get a complexity measure of 3, while a message that consists of 1459 nodes would get a measure of 4. For most analytic purposes, this provides a reasonable value.

- ❑ Traces use the bulk of time and memory in iWay Service Manager. For valid statistics, turn off all traces. You can use the **set trace off** command from the dox box to do this, or you can use the iWay Service Manager Administration Console.

Many books are available on queuing theory, the use of available statistics, and the interpretation of displayed fields.

Kushner, Harold J.; *Heavy Traffic Analysis of Controlled Queuing and Communications Networks*. New York, Springer; (June 8, 2001).

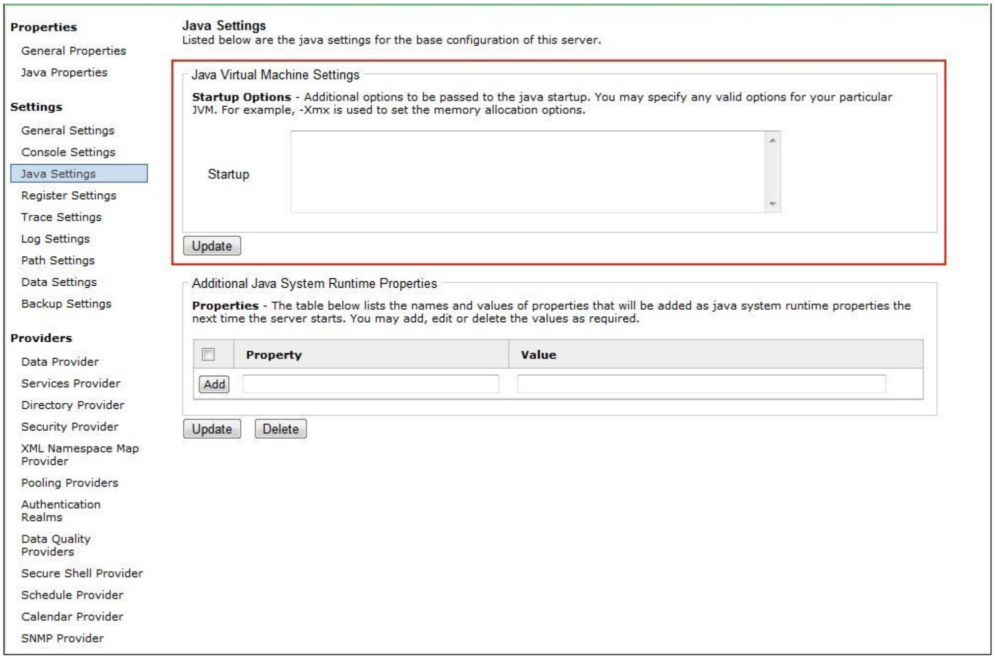
Using JConsole to Monitor iWay Service Manager

The Java Monitoring and Management Console (JConsole) can be used to provide information on iWay Service Manager (iSM) performance and resource consumption running on a Java platform. The JConsole uses Java Management Extension (JMX) technology.

Procedure: How to Use JConsole to Monitor iSM

1. Log on to the iWay Service Manager Administration Console.
2. Click *Java Settings* in the left pane.

3. In the Java Virtual Machine Settings section, specify the Java options from the GUI configuration.



Java Settings

Listed below are the java settings for the base configuration of this server.

Java Virtual Machine Settings

Startup Options - Additional options to be passed to the java startup. You may specify any valid options for your particular JVM. For example, -Xmx is used to set the memory allocation options.

Additional Java System Runtime Properties

Properties - The table below lists the names and values of properties that will be added as java system runtime properties the next time the server starts. You may add, edit or delete the values as required.

4. Click *Update*.

   Note: Use caution when modifying the Java settings. If you make a mistake, then iSM will not start. Perform the following steps if you encounter a problem.

   a. Navigate to the following directory:

      iwayhome\config

      where:

      iwayhome

      Is the name of the directory where iSM is installed.

   b. Edit the config.xml file.

      The config.xml file contains all of the parameters that you added in the Java Virtual Machine Settings section of the iSM console.

      The value in this file is as follows:

      ```
      -Dcom.sun.management.jmxremote.port=12345
      ```

You can use the following example for reference purposes when using a remote system for access:

```
-Dcom.sun.management.jmxremote.port=12345
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false
```

c. Stop and start iSM.

d. Using JConsole, access iSM by entering the host name and port.

Testing Functions

The testfuncs tool enables entry of an expression in the iWay functional language. It evaluates the expression and returns the result. It is intended for technical users.

For more information on using the *testfuncs* command, see the *iWay Service Manager Command Reference Guide*.

To use this command, type:

```
Tool testfuncs <path to an xml document>
```

The tool supports the set subcommand to set a special register value. For example, assume you want to try how the special register works with arithmetic:

```
funcs->set aa 1
stored
funcs->sreg(aa)+2
<superroot [baseNode]>
   <arith [funcNodeMath]>+
      <sreg [funcNodeFunctionSreg]>sreg(aa)
         <p-literal [funcNodeLit]>aa</p-literal>
      </sreg>
      <x-literal [funcNodeLit]>2</x-literal>
   </arith>
</superroot>
3
funcs->
```

Another example might make it clearer:

```
funcs->_substr('abcde',2,4)
<superroot [baseNode]>
   <substr [funcNodeFunctionStr.substr]>_substr(&apos;abcde&apos;,2,4)
      <p-literal [funcNodeLit]>abcde</p-literal>
      <p-literal [funcNodeLit]>2</p-literal>
      <p-literal [funcNodeLit]>4</p-literal>
   </substr>
</superroot>
cd
funcs->
```

Each test shows the abstract syntax tree that results from the compile of the function. Problems with the compilation can usually be understood by analysis of the tree. Some function testing requires use of special registers. The following command sets the specified special register to the designated value.

```
set regname value
```

The value operand is evaluated, so that a value can, for example, reside in a file. If the value after evaluation begins with a left bracket, the test tool assumes that the value is to be parsed as XML and an XML tree is to be loaded into the named register. Otherwise, the register is set to a string of the input value. To set register one to the value valone, use:

```
funcs->set one valone
stored
```

A file in the root named sregdoc.xml contains an XML document. To load it into a special register named xmldoc, use the following command.

```
funcs->set xmldoc file('/sregdoc.xml')
stored xml
```

Testing XPATH

The testxpath tool enables you to try xpath expressions against a standard document. The xpath expression will be evaluated against the provided document and the result returned.

For more information on using the *testxpath* command, see the *iWay Service Manager Command Reference Guide*.

To use this command, type:

```
Tool testxpath <sampledocument>
```

Assume the standard document is the same as that shown in testfuncs.

```
Enter command:>tool testxpath \smalldoc.xml
<?xml version="1.0" encoding="ISO-8859-1" ?>
<a>
   <top>
      <b>one</b>
      <b>two</b>
      <x>sreg(iwayhome)</x>
   </top>
</a>
```

Now enter an xpath expression against the sample document.

```
xpath->/a/top/b[1]
<superRoot [baseNode]>
   <a [baseNode]/>
   <top [baseNode]/>
   <b [baseNode]>
      <functionPredicate [filterInt]>1</functionPredicate>
   </b>
</superRoot>
values->'one'
tree-><?xml version="1.0" encoding="ISO-8859-1" ?>
<xpathresult>
   <b>one</b>
</xpathresult>
list-><b>one [parent=top]
xpath->
```

Using Debug Nodes

The process flow designer provides Debug objects that can be inserted into a process flow. Debug objects wrap the QA Service (com.ibi.agents.XDQAAgent), which displays the contents of registers and the current document, along with useful information about the document (for example, number of siblings, error mode, and so on).

Debug nodes are active in process flow tests from iIT if the process flow itself is running in debug mode. This mode is set as a property of the process flow itself. If the flow is not running in debug mode, then the nodes are ignored.

Normally, a process flow running in an actual channel has the debug nodes disabled. The following command controls the use of the debug nodes in a channel:

```
set pflowdebug on | off [-m <channelname>]
```

The command can activate the debug nodes for the entire iSM or for a single channel.

For more information on using the *pflowdebug* command, see the *iWay Service Manager Command Reference Guide*.

For more information on using the QA Service (com.ibi.agents.XDQAAgent), see the *iWay Service Manager Component Reference Guide*.

Using the Log Viewer

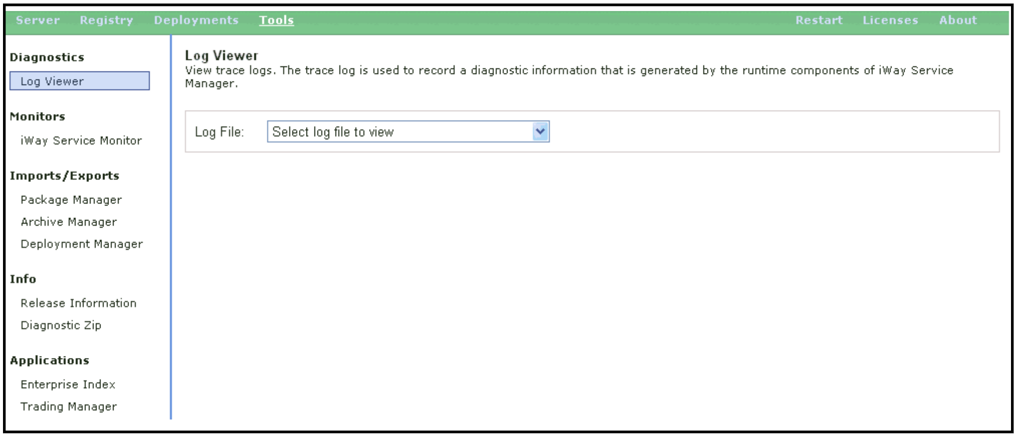
The Log Viewer manages the display properties of system debugging information when the logging and tracing functions are activated. It filters and displays debugging information as each transaction is received and processed. The Log Viewer also displays the date/time range, type, source, and message of every trace entry.

Note: In order to display traces of a specific level, you must have previously enabled them to be written to the log file. For more information, see *Log Settings* on page 475 and *Trace Settings* on page 477.

Procedure: How to Use the Log Viewer

1. Click *Tools* in the top pane and select *Log Viewer* from the Diagnostics section in the left pane.

   The Log Viewer pane opens, as shown in the following image.



Log Viewer

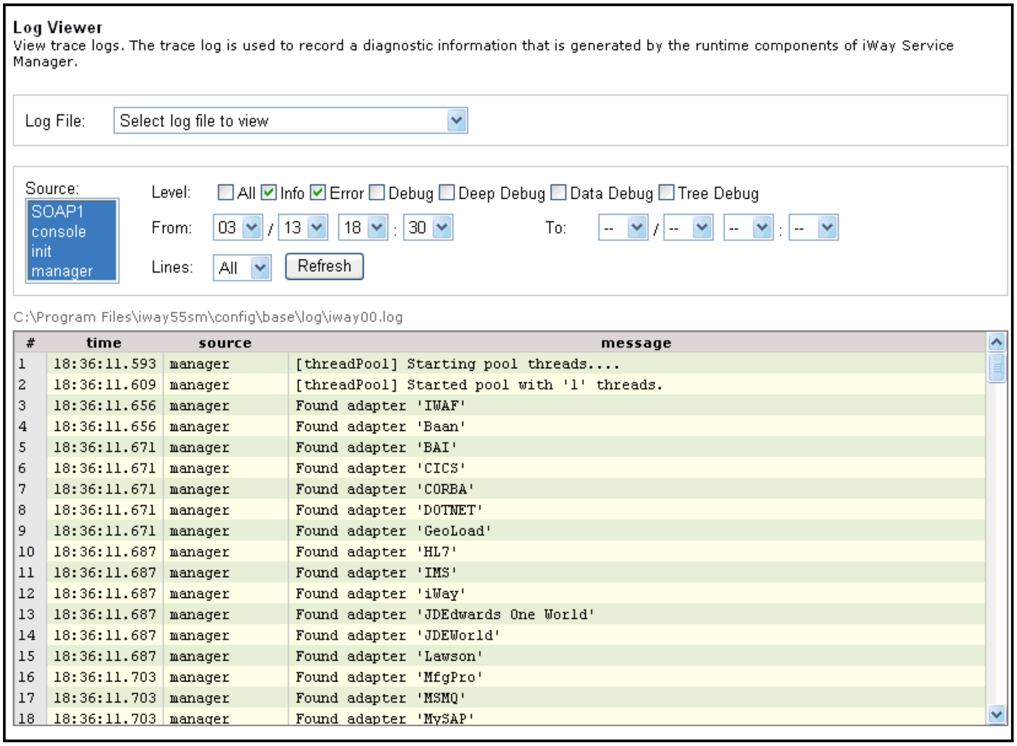
View trace logs. The trace log is used to record a diagnostic information that is generated by the runtime components of iWay Service Manager.

Log File: Select log file to view

2. Select a specific log file to view from the Log File drop-down list.

   Note: Log file names are reused in a circular queue so that they will not proliferate and consume too much disk space. The date/time stamp is shown in the drop-down list in order to show the correct sequence of the files.

The Log Viewer pane is automatically refreshed and shows the log file you selected.



Log Viewer

View trace logs. The trace log is used to record a diagnostic information that is generated by the runtime components of iWay Service Manager.

Log File: Select log file to view

Source: SOAP1, console, init, manager

Level: □ All ☑ Info ☑ Error □ Debug □ Deep Debug □ Data Debug □ Tree Debug

From: 03 / 13 18 : 30 To: -- / -- -- : --

Lines: All Refresh

C:\Program Files\iway55sm\config\base\log\iway00.log

#	time	source	message
1	18:36:11.593	manager	[threadPool] Starting pool threads....
2	18:36:11.609	manager	[threadPool] Started pool with '1' threads.
3	18:36:11.656	manager	Found adapter 'IWAF'
4	18:36:11.656	manager	Found adapter 'Baan'
5	18:36:11.671	manager	Found adapter 'BAI'
6	18:36:11.671	manager	Found adapter 'CICS'
7	18:36:11.671	manager	Found adapter 'CORBA'
8	18:36:11.671	manager	Found adapter 'DOTNET'
9	18:36:11.671	manager	Found adapter 'GeoLoad'
10	18:36:11.687	manager	Found adapter 'HL7'
11	18:36:11.687	manager	Found adapter 'IMS'
12	18:36:11.687	manager	Found adapter 'iWay'
13	18:36:11.687	manager	Found adapter 'JDEdwards One World'
14	18:36:11.687	manager	Found adapter 'JDEWorld'
15	18:36:11.687	manager	Found adapter 'Lawson'
16	18:36:11.703	manager	Found adapter 'MfgPro'
17	18:36:11.703	manager	Found adapter 'MSMQ'
18	18:36:11.703	manager	Found adapter 'MySAP'

3. Select the source component, level, date/time range, and number of lines to display and click *Refresh*.

   The contents of the log file, as filtered by your criteria, are displayed.

   Tip: Multiple trace sources can be selected by using *Ctrl + click*.

Creating a Diagnostic Zip

The Create Diagnostic Zip option provides a quick way to collect the current configuration and log files. An iWay Software support representative may ask you to create a diagnostic zip for problem analysis.

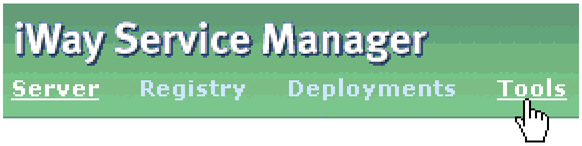
You can use this function to add any relevant comments to the file. The file is labeled with a timestamp in your configuration directory.

Note: Remove previous trace files prior to running a diagnostic zip.

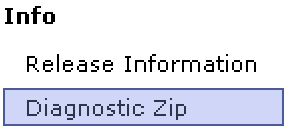
For information on including and excluding files from the class path, see *Operations and Monitoring* on page 389.

Procedure: How to Create a Diagnostic Zip

To create a diagnostic zip:

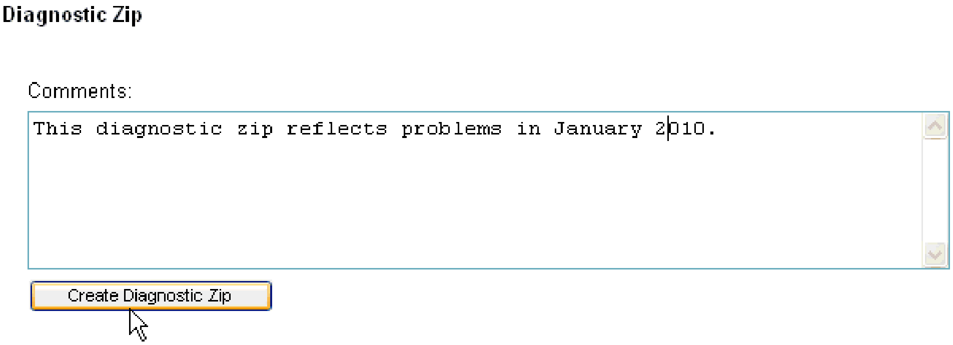


1. Click *Tools* in the menu bar, which is located in the top pane.



2. In the left pane, select *Diagnostic Zip*.

   The Diagnostic Zip pane opens, as shown in the following image.

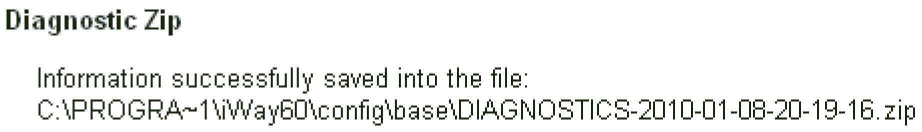


Comments:

This diagnostic zip reflects problems in January 2010.

3. Type your comments in the space that is provided.
4. Click *Create Diagnostic Zip*.

In this example, if you are using the base configuration, the file is saved to the location shown in the following image.



Diagnostic Zip

Information successfully saved into the file:
C:\PROGRA~1\iWay60\config\base\DIAGNOSTICS-2010-01-08-20-19-16.zip

Appendix A

iSM Special Registers (SREGs)

This appendix lists and describes some of the Special Registers (SREGs) in iWay Service Manager (iSM).

In this appendix:

Global SREGs

This section lists and describes some of the global Special Registers (SREGs) in iWay Service Manager (iSM). These SREGs are read-only and cannot be set in a process flow.

SREGs are created and presented to the application as needed. In addition to the SREGs established to describe iSM settings, each listener in a channel Inlet sets SREGs as appropriate to the protocol (for example, the File listener creates *sreg(filename)* to show the name of the file being processed. The listener SREGs are shown in the iSM Administration Console when a listener of the protocol type is defined.

SREGs created by individual services/agents are described in the *iWay Service Manager Component Reference Guide* where the specific service/agent is documented.

The SREGs and their values can be seen at any point in a process flow by use of the QA Service (com.ibi.agents.XDQAAgent) or Debug Node in the process flow.

engine

For example:

```
sreg(engine) = 'base'
```

ibse-port

For example:

```
sreg(ibse-port) = '9000'
```

iway.pid

For example:

```
sreg(iway.pid) = '7848'
```

iway.serverfullhost

For example:

```
sreg(iway.serverfullhost) = 'iwayntk1.ibi.com'
```

iway.serverhost

For example:

```
sreg(iway.serverhost) = 'iwayntk1'
```

iway.serverip

For example:

```
sreg(iway.serverip) = '172.30.173.25'
```

iwayconfig or iway.config

For example:

```
sreg(iwayconfig) = 'base'
```

iwayhome

For example:

```
sreg(iwayhome) = 'C:/temp/iway7/'
```

iwayworkdir or iway.workdir

For example:

```
sreg(iwayworkdir) = 'C:/temp/iway7/config/base'
```

iwayversion

For example:

```
sreg(iwayversion) = '7.0.4'
```

iway.startup.time

For example:

```
sreg(iway.startup.time) = 1429477701703
```

Transform SREGs

This section lists and describes the transform Special Registers (SREGs) in iWay Service Manager (iSM).

iway.transform.errormask

A global setting for error masking that can be used in the event that it is required to permanently suppress non-fatal errors in the run-time log for all transformations running on the specific format. This SREG can be set for masking input and/or output transform error levels. Define and set the SREG to *true* only if you wish to suppress all input (and/or) output non-fatal levels in the log for all transformations using the specific format.

For example:

```
iway.transform.errormask.<phase>.<format>
```

where:

<phase>

Can be set as *in* or *out*.

<format>

Is any data format supported by iWay Transformer.

For instance, if you would like to mask all non-fatal errors for Fixed Width Format for input and output, then define and set to *true* the following two global SREGs:

```
setsreg(‘iway.transform.errormask.in.fwf’, ‘true’)

setsreg(‘iway.transform.errormask.out.fwf’, ‘true’)
```

For more information on setting SREG values, see the *iWay Functional Language Reference Guide*.

For more information on the available data formats supported by the Transformer tool, see the *iWay Integration Tools Transformer User’s Guide*.

Process Flow SREGs

This section lists and describes the process flow Special Registers (SREGs) in iWay Service Manager (iSM). These SREG values are read-only and will change with the process and channel where they are referenced.

iway.flowname

This SREG contains the name of the process flow that is currently executing. For example:

```
sreg(iway.flowname) = 'PF_0125_flow_2'
```

iway.channel

This SREG contains the name of the channel that is currently executing. For example:

```
sreg(iway.channel) = 'Channel_File1'
```

Legal and Third-Party Notices

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the readme file for the availability of this software version on a specific operating system platform.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.